Curry-style type isomorphisms and game semantics

Joachim de Lataillade
Preuves Programmes Systèmes
CNRS - Paris 7
Joachim.de-Lataillade@pps.jussieu.fr

Abstract

Curry-style system F, i.e. system F with no explicit types in terms, can be seen as a core presentation of polymorphism from the point of view of programming languages.

This paper gives a characterisation of type isomorphisms for this language, by using a game model whose intuition comes both from the syntax and from the game semantics universe. The model is composed of: an untyped part to interpret terms, a notion of game to interpret types, and a typed part to express the fact that an untyped strategy σ plays on a game A.

By analysing isomorphisms in the model, we prove that the equational system corresponding to type isomorphisms for Curry-style system F is the extension of the equational system for Church-style isomorphisms with a new, non-trivial equation: $\forall X.A \simeq_\varepsilon A[\forall Y.Y/X]$ if X appears only positively in A.

1 Introduction

Types isomorphisms. The problem of type isomorphisms is a purely syntactical question: two types A and B are isomorphic if there exist two terms $f : A \to B$ and $g : B \to A$ such that $f \circ g = id_B$ and $g \circ f = id_A$. This equivalence relation on data types allows to translate a program from one type to the other without any change on the calculatory meaning of the program. Thus, a search in a library up to type isomorphism will help the programmer to find all the functions that can potentially serve his purpose, and to reuse them in the new typing context [Rit91]. This is particularly appealing with functional languages, because in this case the type can really be seen as a partial specification of the program: such a library search up to isomorphisms has been implemented in particular for Caml Light by Jérôme Vouillon. It can also be used in proof assistants to help finding proofs in libraries and reusing them [BP01] (for more details on the use of type isomorphisms in computer science, see [DC95]). From a more general point of view, type isomorphisms are the natural answer to the question of equivalence between types in a programming language.

The question of characterising these type isomorphisms is then a very simple problem to formulate, however its resolution is often non-trivial, especially when dealing with polymorphism. Roberto Di Cosmo [DC95] has solved syntactically this question for *Church-style* system F (i.e. system F where types appear explicitly in the terms) by giving an equational system on types equivalent to type isomorphisms. In a preceding work [dL07], we have given a new proof of this result by using a game semantics model of Church-style system F. In this more geometrical approach, types were interpreted by an arborescent structure, *hyperforests*: the natural equality for this structure happened to be exactly the equality induced by type isomorphisms. The efficiency of game semantics in this context was an incitement to go further and to explore the possibility of resolving this question for other languages.

Curry-style system F. In the present work, we deal with type isomorphisms for *Curry-style* system F, i.e. system F where the terms grammar is simply the untyped λ-calculus' one. Although this system appears to be less relevant than Church-style system F in proof-theory (a term does not correspond exactly to one proof), it is actually more accurate when we consider programming languages. Indeed, in Church-style system F, a term t of type $\forall X.A$ will not have the type $A[B/X]$: only $t\{B\}$ will be of this type; whereas in Curry-style, a term t of type $\forall X.A$ will have all the types $A[B/X]$, which is more the idea induced by the notion of polymorphism: the same function may be used with different types. The typing rules and equalities of this language are presented on figure 1.

Grammars:

$$\begin{array}{rl} A ::= & X \mid A \to A \mid \forall X.A \mid A \times A \mid \bot \\ t ::= & x \mid \lambda x.t \mid (tt) \mid \langle t, t\rangle \mid \pi_1(t) \mid \pi_2(t) \end{array}$$

Typing rules:

$$\frac{}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i}\,(\mathrm{ax})$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\,(\to I)$$

$$\frac{\Gamma \vdash t : A \to B \qquad \Gamma \vdash u : A}{\Gamma \vdash (tu) : B}\,(\to E)$$

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : B}{\Gamma \vdash \langle t, u\rangle : A \times B}\,(\times I)$$

$$\frac{\Gamma \vdash t : A \times B}{\Gamma \vdash \pi_1(t) : A}\,(\times E1) \qquad \frac{\Gamma \vdash t : A \times B}{\Gamma \vdash \pi_2(t) : B}\,(\times E2)$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall X.A}\,(\forall I)\ \text{ if } X \notin \Gamma$$

$$\frac{\Gamma \vdash t : \forall X.A}{\Gamma \vdash t : A[B/X]}\,(\forall E)$$

Equalities:

$$\begin{array}{rclll} (\lambda x.t)u & = & t[u/x] & & (\beta) \\ \lambda x.tx & = & t & \text{if } x \notin t & (\eta) \\ \pi_1(\langle u, v\rangle) & = & u & & (\pi_1) \\ \pi_2(\langle u, v\rangle) & = & v & & (\pi_2) \\ \langle \pi_1(u), \pi_2(u)\rangle & = & u & & (\times) \end{array}$$

Type isomorphism:

$$(t, u) \text{ s.t. } \begin{cases} \vdash t : A \to B \\ \vdash u : B \to A \\ \lambda x.t(ux) = \lambda x.u(tx) = \lambda x.x \end{cases}$$

Figure 1: Curry-style system F

Compared with this system, Church-style system F has a different grammar of terms:

$$t ::= x \mid \lambda x^A.t \mid (tt) \mid \langle t,t\rangle \mid \pi_1(t) \mid \pi_2(t) \mid \ \Lambda X.t \mid \ t\{A\}$$

different typing rules for the quantification:

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \Lambda X.t : \forall X.A}(\forall I) \ \text{ if } X \notin \Gamma \qquad \frac{\Gamma \vdash t : \forall X.A}{\Gamma \vdash t\{B\} : A[B/X]}(\forall E)$$

and two additional equalities:

$$\begin{array}{rcll} (\Lambda X.t)\{A\} & = & t[A/X] & & (\beta 2) \\ \Lambda X.t\{X\} & = & t & \text{if } X \notin t & (\eta 2) \end{array}$$

As can be seen on the typing rules, a λ-term t is of type A if there exists a term $\tilde{t}$ of Church-style system F such that t is obtained from $\tilde{t}$ by erasing all the type indications (for example, $\Lambda X.\lambda x^{\forall Y.Y}\lambda y^Y.x\{Y\}$ becomes $\lambda x \lambda y.x$). In this case, we say that t is the *erasure* of $\tilde{t}$.

The characterisation of type isomorphisms for Curry-style system F is not directly reducible to the Church-style corresponding question: indeed, types of the form $\forall X.A$ and A with $X \notin A$ are not equivalent in the Church-style setting, but they are in the Curry-style one (where the isomorphism is realised by the identity). We prove in this paper that the distinction between Church-style and Curry-style type isomorphisms can be resumed in one new and non-trivial equation. To express it, one first have to recall the definition of positive and negative type variables in a formula[1]:

Definition 1 *If A is a formula, its sets of* ***positive variables*** *Pos_A and* ***negative variables*** *Neg_A are defined by:*

- $Pos_X = \{X\}\ ,\ Neg_X = \emptyset$
- $Pos_\perp = Neg_\perp = \emptyset$
- $Pos_{A\times B} = Pos_A \cup Pos_B\ ,\ Neg_{A\times B} = Neg_A \cup Neg_B$
- $Pos_{A\to B} = Neg_A \cup Pos_B\ ,\ Neg_{A\to B} = Pos_A \cup Neg_B$
- $Pos_{\forall X.A} = Pos_A \setminus \{X\}\ ,\ Neg_{\forall X.A} = Neg_A \setminus \{X\}$

We also define $FTV(A) = Pos_A \cup Neg_A$.

The new equation is then the following:

$$\forall X.A \simeq_\varepsilon A[\forall Y.Y/X] \qquad \text{if } X \notin Neg_A$$

It is true in Curry-style but false (in general) in Church-style system F. Note that, although the isomorphism is realised by the identity, the Church-style terms $t : \forall X.A \to A[\forall Y.Y/X]$ and $u : A[\forall Y.Y/X] \to \forall X.A$, from which we extract the identity by erasing explicit types, are not trivial (they will be explicitly described in the proof of theorem 2 at the end of the paper). This is a difference with Church-style system F, where type isomorphisms were exactly the expected ones, even if proving that point was not an elementary task.

Type isomorphisms for Curry-style system F are finally characterised by the following equational system:

$$\begin{array}{rcll} A \times B & \simeq_\varepsilon & B \times A & \\ A \times (B \times C) & \simeq_\varepsilon & (A \times B) \times C & \\ A \to (B \to C) & \simeq_\varepsilon & (A \times B) \to C & \\ A \to (B \times C) & \simeq_\varepsilon & (A \to B) \times (A \to C) & \\ \forall X.\forall Y.A & \simeq_\varepsilon & \forall Y.\forall X.A & \\ A \to \forall X.B & \simeq_\varepsilon & \forall X.(A \to B) & \text{if } X \notin FTV(A) \\ \forall X.(A \times B) & \simeq_\varepsilon & \forall X.A \times \forall X.B & \\ \forall X.A & \simeq_\varepsilon & A[\forall Y.Y/X] & \text{if } X \notin Neg_A \end{array}$$

[1] All along this article we will identify the notions of *type* and *formula* (according to the Curry-Howard correspondence).

The purpose of this paper is to prove correctness and completeness of this characterisation by using a game model.

The model. Models of second order calculi do not come about easily due to impredicativity. Among the different possibilities, we choose models based on game semantics because of their high degree of adequation with the syntax: indeed, game semantics has been widely used to construct fully complete models for various calculi, such as PCF [AJM00, HO00], μPCF [Lai97], Idealized Algol [AM99], etc. This means that this semantics gives a very faithful description of the behaviour of the syntax modulo reduction rules in the system. And this is precisely what we need to deal semantically with type isomorphisms: a model which is so precise that it contains no more isomorphisms than the syntax.

The present paper introduces a game model for Curry-style system F. This model was largely inspired by two preceding game semantics works: the PhD thesis of Juliusz Chroboczek [Chr03], which presents among others a game semantics for an untyped calculus that we will almost copy-paste in this paper; and the game semantics model for generic polymorphism by Samson Abramsky and Radha Jagadeesan [AJ03], from which we will extract many ideas in our context. Other game semantics models had an influence on our work: Dominic Hughes gave the first game models of Church-style system F [Hug00] and introduced the notion of *hyperforests* that we reuse here; Andrzej Murawski and Luke Ong presented a simple and efficient model for dealing with affine polymorphism [MO01], and their presentation of moves inspired ours.

It shall be noticed that the design of our Curry-style game model is actually very connected to the concepts present in the syntax: the notion of erasure we introduce is of course reminiscent of the erasure of types in a Church-like term to obtain a Curry-like term. This is no surprise as we need a model describing very precisely the syntax (that is why, in particular, one cannot be satisfied by an interpretation of the quantification as an intersection or a greatest lower bound). The specificities of (HON-)game semantics, as for example the arborescent structure that interprets types, are however decisive for our demonstration.

2 General definitions

In this section we give general constructions that will apply on the different grammars we use in the model. These constructions are strongly related to usual HON-style games operations (cf. [HO00]).

2.1 Moves

We consider the set of type variables $X, Y, \ldots$ to be in bijection with $\mathbb{N}\backslash\{0\}$, and we will further write this set $\mathcal{X} = \{X_j \mid j > 0\}$.

All along this article, we define several grammars of the form:

$$\mu ::= \uparrow\mu \mid \downarrow\mu \mid \alpha_i\mu \mid j \qquad (i \in I,\ j \in \mathbb{N})$$

Let us note $\mathcal{M}$ the set of words (often called **moves**) defined by this grammar.

Intuitively, the token $\uparrow$ (resp. $\downarrow$) corresponds to the right side (resp. the left side) of an arrow type, the α_i's are related to additional (covariant) connectors, the constants $j \in \mathbb{N}\backslash\{0\}$ correspond to free type variables X_j and the constant 0 corresponds either to bounded type variables or to $\perp$.

On such a grammar, we define automatically a function λ of **polarity**, with values in $\{\mathbf{O}, \mathbf{P}\}$:

- $\lambda(j) = \mathbf{O}$
- $\lambda(\uparrow\mu) = \lambda(\alpha_i\mu) = \lambda(\mu)$
- $\lambda(\downarrow\mu) = \overline{\lambda}(\mu)$

where $\overline{\mathbf{O}} = \mathbf{P}$ and $\overline{\mathbf{P}} = \mathbf{O}$.

We also introduce an **enabling relation** $\vdash \subseteq \mathcal{M} \cup (\mathcal{M} \times \mathcal{M})$:

- $\vdash j$
- if $\vdash \mu$ then $\vdash \alpha_i\mu$, and $\vdash \uparrow\mu$
- if $\vdash \mu$ and $\vdash \mu'$ then $\uparrow\mu \vdash \downarrow\mu'$
- if $\mu \vdash \mu'$ then $\alpha_i\mu \vdash \alpha_i\mu'$, $\uparrow\mu \vdash \uparrow\mu'$ and $\downarrow\mu \vdash \downarrow\mu'$.

which induces a partial order $\leq$ for this grammar by reflexive and transitive closure. If $\vdash \mu$ we say that μ is an **initial move** (in which case $\lambda(\mu) = \mathbf{O}$).

2.2 Substitution

As we want to deal with polymorphism, we need some operations acting directly on the leafs j:

- a function $\sharp$ of **leaf extracting**:
  - $\sharp(j) = j$
  - $\sharp(\uparrow\mu) = \sharp(\downarrow\mu) = \sharp(\alpha_i\mu) = \sharp(\mu)$
- an operation of **substitution** $\mu[\mu']$:
  - $j[\mu'] = \mu'$
  - $\uparrow\mu[\mu'] = \uparrow(\mu[\mu'])$, $\downarrow\mu[\mu'] = \downarrow(\mu[\mu'])$ and $\alpha_i\mu[\mu'] = \alpha_i(\mu[\mu'])$

We say that μ_1 is a **prefix** of μ_2 if there exists $\mu' \in \mathcal{M}$ such that $\mu_2 = \mu_1[\mu']$. This is denoted $\mu_1 \sqsubseteq^p \mu_2$.

2.3 Plays and strategies

Definition 2 (justified sequence, play) *A* ***justified sequence*** *on a given grammar is a sequence $s = \mu_1 \dots \mu_n$ of moves, together with a partial function $f : \{1, \dots, n\} \rightharpoonup \{1, \dots, n\}$ such that: if $f(i)$ is not defined then $\vdash \mu_i$, and if $f(i) = j$ then $j < i$ and $\mu_j \vdash \mu_i$: in this case we say that μ_j* ***justifies*** *μ_i.*

A ***play*** *on a grammar is a justified sequence $s = \mu_1 \dots \mu_n$ on this grammar such that: for every $1 \leq i \leq n-1$, if $\lambda(\mu_i) = \mathbf{P}$ then $\lambda(\mu_{i+1}) = \mathbf{O}$ and if $\lambda(\mu_i) = \mathbf{O}$ then $\lambda(\mu_{i+1}) = \mathbf{P}$ and $\sharp(\mu_i) = \sharp(\mu_{i+1})$.*

We note $\mathbb{E}$ the set of plays of even length. If s and t are two plays, we note $t \preceq s$ if t is a prefix of s.

The definition of a play implies that if $s\mu\nu$ is an even-length play then $\sharp(\mu) = \sharp(\nu)$. This will be a very significant property in our model.

Definition 3 (strategy) *A* ***strategy*** *σ on a given grammar is a non-empty set of even-length plays, which is closed under even-length prefix and deterministic: if $s\mu$ and $s\nu$ are two plays of σ then $s\mu = s\nu$.*

Definition 4 (view, innocence) *Let s be a play on a grammar, we define its* ***view*** *$\ulcorner s \urcorner$ by:*

- $\ulcorner \varepsilon \urcorner = \varepsilon$
- $\ulcorner s\mu \urcorner = \ulcorner s \urcorner \mu$ *if* $\lambda(\mu) = \mathbf{P}$
- $\ulcorner s\mu \urcorner = \mu$ *if* $\vdash \mu$
- $\ulcorner s\mu t\nu \urcorner = \ulcorner s \urcorner \mu\nu$ *if* $\lambda(\nu) = \mathbf{O}$ *and μ justifies ν*

A strategy σ is called ***innocent*** *if, for every play $s\nu$ of σ, the justifier of ν is in $\ulcorner s \urcorner$, and if we have: if $s\mu\nu \in \sigma$, $t \in \sigma$, $t\mu$ is a play and $\ulcorner s\mu \urcorner = \ulcorner t\mu \urcorner$ then $t\mu\nu \in \sigma$.*

Definition 5 (bi-view) *A* ***bi-view*** *on a given grammar is a justified sequence $s = \mu_1 \dots \mu_n$ (with $n \geq 1$) such that any move is justified by its predecessor. The set of bi-views is denoted $\mathcal{BV}$.*

2.4 Composition

Composition is usually defined between arenas of the form $A \to B$ and $B \to C$. We are going to define it in a context where arenas do not explicitly exist, but are however represented by the tokens $\uparrow$ and $\downarrow$.

Definition 6 (shape) *Let $\zeta \in (\{\uparrow, \downarrow\} \cup \{\alpha_i\}_{i \in I})^*$, a move μ is said to be of **shape** ζ if $\zeta 0 \sqsubseteq^p \mu$.*

Let Σ be a finite set of elements $\zeta_j \in (\{\uparrow, \downarrow\} \cup \{\alpha_i\}_{i \in I})^*$. A justified sequence is said to be of shape Σ if each of its moves is of shape ζ_j for some j. A strategy is of shape Σ if each of its plays is of shape Σ.

*In the case where $\Sigma = \{\uparrow, \downarrow\}$, we say that the justified sequence (or the strategy) is of **arrow shape**.*

Consider a justified sequence $s = \mu_1 \dots \mu_n$, we define the sequence $s \upharpoonright_\zeta$ as the restriction of s to the moves of shape ζ where the prefix ζ has been erased, and the pointers are given as follows: if $\mu_i = \zeta \mu'_i$ is justified by $\mu_j = \zeta \mu'_j$ in s, then the corresponding occurrence of μ'_i is justified by μ'_j

Consider $\zeta, \xi \in (\{\uparrow, \downarrow\} \cup \{\alpha_i\}_{i \in I})^*$ such that neither of the two is a prefix of the other. Let us define the sequence $s \upharpoonright_{\zeta, \xi}$: first we consider s', the restriction of s to the moves of shape ζ and the moves of shape ξ hereditarily justified by a move of shape ζ. $s \upharpoonright_{\zeta, \xi}$ is the sequence s' where the prefix ζ has been replaced by $\uparrow$ where it appears, the prefix ξ has been replaced by $\downarrow$ where it appears, and the pointers are given as follows: if $\mu_i = \zeta \mu'_i$ (resp. $\mu_i = \xi \mu'_i$) is justified by $\mu_j = \zeta \mu'_j$ (resp. $\mu_j = \xi \mu'_j$) in s, then the corresponding occurrence of $\uparrow \mu'_i$ (resp. $\downarrow \mu'_i$) is justified by $\uparrow \mu'_j$ (resp. $\downarrow \mu'_j$); and if $\mu_i = \xi \mu'_i$ is hereditarily justified by a move $\mu_j = \zeta \mu'_j$ in s, then the corresponding occurrence of $\downarrow \mu'_i$ is justified by the corresponding occurrence of $\uparrow \mu'_j$ iff $\vdash \mu'_i$ and $\vdash \mu'_j$.

Definition 7 (interacting sequence, composition) *An **interacting sequence** $s = \mu_1 \dots \mu_n$ is a justified sequence of shape $\{\uparrow, \downarrow\uparrow, \downarrow\downarrow\}$ such that $s \upharpoonright_{\uparrow, \downarrow\uparrow}$, $s \upharpoonright_{\downarrow\uparrow, \downarrow\downarrow}$ and $s \upharpoonright_{\uparrow, \downarrow\downarrow}$ are plays. The set of interacting sequences is denoted **Int**.*

*Suppose we have two strategies σ and τ. We call **composition** of σ and τ the set of plays*

$$\sigma; \tau = \{u \upharpoonright_{\uparrow, \downarrow\downarrow} \mid u \in \textbf{\textit{Int}},\ u \upharpoonright_{\uparrow, \downarrow\uparrow} \in \tau \text{ and } u \upharpoonright_{\downarrow\uparrow, \downarrow\downarrow} \in \sigma\}$$

$\sigma; \tau$ is a strategy: this can be proven like in the standard HON game model. Moreover if σ and τ are innocent then $\sigma; \tau$ is innocent.

Definition 8 (totality on a shape) *Let σ be a strategy and $\zeta \in (\{\uparrow, \downarrow\} \cup \{\alpha_i\}_{i \in I})^*$. We say that σ is **total** on the shape ζ if, for every play $s \in \sigma$ of shape ζ, for every move μ such that $s\mu$ is a play of shape ζ, there exists a move ν of shape ζ such that $s\mu\nu \in \sigma$.*

2.5 Presentation of the Curry-style model

Our model is defined through three grammars:

- $\mathbb{X}$ is the grammar of **untyped moves** which generate the untyped model to interpret untyped lambda-terms
- $\mathbb{A}$ is the grammar of **occurrences** which are used for the interpretation of formulas
- $\mathbb{M}$ is the grammar of **typed moves** which generate an interpretation of the terms of Church-style system F.

The interpretation of Curry-style system F in the model will be as follows:

- a type A will be interpreted as a *game* (also denoted A), i.e. a specific structure based on the grammar $\mathbb{A}$

- a term t of type A will be interpreted as a strategy σ on the grammar $\mathbb{X}$, with the condition that this strategy is the **erasure** of a strategy $\tilde{\sigma}$, defined on the grammar $\mathbb{M}$ and played on the game A (this will be denoted $\tilde{\sigma} :: A$)
- two additional properties are required: **hyperuniformity** which applies on σ, and **uniformity** which applies on $\tilde{\sigma}$.

In what follows, we first define the untyped model to interpret untyped lambda-terms, then we define games and typed strategies on games, and finally we introduce the notion of erasure and prove that we have a model of Curry-style system F. Next we prove, using this model, our result on type isomorphisms.

3 The untyped model

In this section we give a semantics for the untyped λ-calculus with binary products, i.e. for the calculus of figure 1 restricted to the language of terms with their reduction rules.

The untyped model that we present below has been defined by Julius Chroboczek in his PhD thesis [Chr03]. Our definition is formally a little bit different from Chroboczek's one, but the substance of the work is the same.

3.1 Untyped moves

The grammar of **untyped moves** is the following:

$$x ::= \uparrow x \mid \downarrow x \mid rx \mid lx \mid j \qquad (j \in \mathbb{N})$$

The set of untyped moves is denoted $\mathbb{X}$.

The justified sequences, plays and strategies induced by this grammar will be called *untyped* justified sequences, plays and strategies.

3.2 Basic strategies

We define the following strategies:

- **identity**:
$$id = \{s \in \mathbb{E} \mid s \text{ of arrow shape and } \forall t \in \mathbb{E}, t \leq s \Rightarrow t\restriction_{\uparrow} = t\restriction_{\downarrow}\}$$
- **projections**:
$$\pi_r = \{s \in \mathbb{E} \mid s \text{ of shape } \{\uparrow, \downarrow r, \downarrow l\} \text{ and } \forall t \in \mathbb{E}, t \leq s \Rightarrow t\restriction_{\uparrow} = t\restriction_{\downarrow r}\}$$
$$\pi_l = \{s \in \mathbb{E} \mid s \text{ of shape } \{\uparrow, \downarrow r, \downarrow l\} \text{ and } \forall t \in \mathbb{E}, t \leq s \Rightarrow t\restriction_{\uparrow} = t\restriction_{\downarrow l}\}$$
- **evaluation**:
$$eval = \{s \in \mathbb{E} \mid s \text{ of shape } \{\uparrow, \downarrow l\uparrow, \downarrow l\downarrow, \downarrow r\} \text{ and } \forall t \in \mathbb{E}, t \leq s \Rightarrow t\restriction_{\uparrow} = t\restriction_{\downarrow l\uparrow} \wedge t\restriction_{\downarrow r} = t\restriction_{\downarrow l\downarrow}\}$$

We also define three basic operations on strategies:

- **pairing without context**: if σ and τ are two strategies,
$$\langle \sigma, \tau \rangle_a = \{s \in \mathbb{E} \mid s \text{ of shape } \{r, l\} \text{ and } s\restriction_l \in \sigma \text{ and } s\restriction_r \in \tau\}$$
- **pairing with context**: if σ and τ are two strategies of arrow shape,
$$\langle \sigma, \tau \rangle_b = \{s \in \mathbb{E} \mid s \text{ of shape } \{\uparrow r, \uparrow l, \downarrow\} \text{ and } s\restriction_{\uparrow l, \downarrow} \in \sigma \text{ and } s\restriction_{\uparrow r, \downarrow} \in \tau\}$$
- **abstraction**: if σ is a strategy of shape $\{\uparrow, \downarrow r, \downarrow l\}$, $\Lambda(\sigma)$ is the strategy of shape $\{\uparrow\uparrow, \uparrow\downarrow, \downarrow\}$ which is deduced from σ by replacing each move $\uparrow x$ by $\uparrow\uparrow x$, each move $\downarrow rx$ by $\uparrow\downarrow x$ and each move $\downarrow lx$ by $\downarrow x$.

3.3 Hyperuniformity

We have enough material to define an untyped model. However, our use of untyped strategies in the Curry-style model forces us to impose new requirements: for example, consider the formula $X_1 \to X_1$. It would be reasonable to think that the innocent strategy σ whose set of views is $\{\varepsilon, \uparrow 1 \cdot \downarrow 1\}$ has this type. However, because we deal with a Curry-style model, any strategy of type $X_1 \to X_1$ should also have the type $\forall X_1.X_1 \to X_1$, and thus $A \to A$ for any A, and should be able to do a copycat between the left and the right side of the arrow.

This is the meaning of the notion of **hyperuniformity** defined below.

Definition 9 (copycat extension of an untyped play) *Let $s = x_1 \dots x_n$ be an untyped play, x_i an **O**-move of s and $v = y_1 \dots y_p \in \mathcal{BV}$. Suppose $s = s_1 x_i x_{i+1} s_2$. The **copycat extension** of s at position i with parameter v is the untyped play $s' = cc^s(i, v)$, defined by :*

- $s' = s_1 x_i[y_1] x_{i+1}[y_1] s_2$ *if* $p = 1$
- $s' = s_1 x_i[y_1] x_{i+1}[y_1] x_{i+1}[y_2] x_i[y_2] \dots x_{i+1}[y_p] x_i[y_p]$ *if p even*
- $s' = s_1 x_i[y_1] x_{i+1}[y_1] x_{i+1}[y_2] x_i[y_2] \dots x_i[y_p] x_{i+1}[y_p]$ *if $p > 1$ and p odd*

Definition 10 (hyperuniform strategy) *An untyped strategy σ is called **hyperuniform** if it is innocent and if, for any play $s \in \sigma$, any copycat extension of s is in σ.*

Lemma 1 *The identity strategy, the projections and the evaluation strategy are hyperuniform. If σ and τ are hyperuniform then $\langle \sigma, \tau \rangle$ and $\Lambda(\sigma)$ are hyperuniform.*

The preceding lemma is straightforward. The interesting case is composition:

Lemma 2 *If σ and τ are hyperuniform then $\sigma; \tau$ is hyperuniform.*

Proof: Let us consider a play $s = x_1 \dots x_p \in \sigma; \tau$, an **O**-move x_i of s and a bi-view $v = y_1 \dots y_q$. We have to prove that $s' = cc^s(i, v)$ belongs to $\sigma; \tau$.

There exists a justified sequence u such that $u\upharpoonright_{\uparrow,\downarrow\downarrow} = s$, $u\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \sigma$ and $u\upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau$. If $u = t_1 x_i b_1 \dots b_q x_{i+1} t_2$, we build a new justified sequence U depending on the value of p :

- if $p = 1$, $U = t_1 x_i[y_1] b_1[y_1] \dots b_q[y_1] x_{i+1}[y_1] t_2$
- if p even,
  $U = t_1 x_i[y_1] b_1[y_1] \dots b_q[y_1] x_{i+1}[y_1] x_{i+1}[y_2] b_q[y_2] \dots b_1[y_2] x_i[y_2] \dots\dots x_{i+1}[y_p] b_q[y_p] \dots b_1[y_p] x_i[y_p]$
- if p odd and $p > 1$,
  $U = t_1 x_i[y_1] b_1[y_1] \dots b_q[y_1] x_{i+1}[y_1] x_{i+1}[y_2] b_q[y_2] \dots b_1[y_2] x_i[y_2] \dots\dots x_i[y_p] b_1[y_p] \dots b_q[y_p] x_{i+1}[y_p]$

We have $U\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \sigma$ and $U\upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau$ by hyperuniformity of σ and τ. So, $U\upharpoonright_{\uparrow,\downarrow\downarrow} = s' \in \sigma; \tau$.

□

3.4 Semantics of the untyped λ-calculus with binary products

We now present the interpretation of the untyped calculus. Instead of directly interpreting terms, we interpret sequents of the form $\Gamma \vdash t$, where t is a term and Γ is simply a list of variables that includes the free variables occurring in t.

The interpretation is as follows:

$$\begin{aligned}
[\![x \vdash x]\!] &= id \\
[\![\Gamma, x \vdash x]\!] &= \pi_r \qquad \text{if } \Gamma \neq \emptyset \\
[\![\Gamma, y \vdash x]\!] &= \pi_l; [\![\Gamma \vdash x]\!] \\
[\![\Gamma \vdash \lambda x.t]\!] &= \Lambda([\![\Gamma, x \vdash t]\!]) \\
[\![\Gamma \vdash (tu)]\!] &= \langle [\![\Gamma \vdash t]\!], [\![\Gamma \vdash u]\!] \rangle_{a(\Gamma)}; eval \\
[\![\Gamma \vdash \langle t, u \rangle]\!] &= \langle [\![\Gamma \vdash t]\!], [\![\Gamma \vdash u]\!] \rangle_{a(\Gamma)} \\
[\![\Gamma \vdash \pi_1(t)]\!] &= [\![\Gamma \vdash t]\!]; \pi_l \\
[\![\Gamma \vdash \pi_2(t)]\!] &= [\![\Gamma \vdash t]\!]; \pi_r
\end{aligned}$$

with $a(\Gamma) = a$ if $\Gamma = \emptyset$ and $a(\Gamma) = b$ otherwise.

From lemmas 1 and 2 we derive:

Lemma 3 *Let t be a term whose free variables are contained in the list Γ, then $[\![\Gamma \vdash t]\!]$ is a hyperuniform strategy.*

Proposition 1 *If two terms t and u are equal up to the equalities of the language, and if all their free variables are contained in the list Γ, then $[\![\Gamma \vdash t]\!] = [\![\Gamma \vdash u]\!]$.*

See [Chr03] for the proof of the equivalent proposition in Chroboczek's setting.

4 Games

4.1 Interpretation of a formula

In this section we introduce the notion of **game**[2], the structure that will interpret Curry-style types. This structure is very similar to the one presented in [AJ03].

We define the following grammar of **occurrences**:

$$a ::= \uparrow a \mid \downarrow a \mid ra \mid la \mid \star a \mid j \qquad (j \in \mathbb{N})$$

The set of all occurrences is denoted $\mathbb{A}$.

We define a translation $\mathcal{E}$ from $\mathbb{A}$ to $\mathbb{X}$: $\mathcal{E}(a)$ is obtained by erasing all the tokens $\star$ in a. Inductively:

- $\mathcal{E}(i) = i$
- $\mathcal{E}(\star a) = \mathcal{E}(a)$
- $\mathcal{E}(\alpha a) = \alpha\mathcal{E}(a)$ if $\alpha \in \{\uparrow, \downarrow, r, l\}$.

The **syntactic tree** of a formula A is a tree with nodes labelled by type connectors $(\to, \times, \forall)$ or integers, edges labelled by the tokens $\uparrow, \downarrow, r, l, \star$, and possibly some arrows linking a leaf to a node. It is defined as follows:

- $T_\bot$ is reduced to a leaf 0
- T_{X_i} is reduced to a leaf i

[2]The denomination *arena* would also fit, but we wanted to stress the fact that our games are not trees like HON-arenas, but just partial orders.

- $T_{A\to B}$ consists in a root $\to$ with the two trees T_A and T_B as sons; the edge between $\to$ and T_A (resp. T_B) is labelled $\downarrow$ (resp. $\uparrow$)
- $T_{A\times B}$ consists in a root $\times$ with the two trees T_A and T_B as sons; the edge between $\times$ and T_A (resp. T_B) is labelled l (resp. r)
- $T_{\forall X_i.A}$ consists in a root $\forall$ with the tree T as unique son, where T is deduced from T_A by linking each of its leafs labelled by i to its root, and relabelling these leafs by 0; the edge between $\forall$ and T is labelled $\star$.

A maximal branch in a syntactic tree is a path from the root to a leaf; it will be described by the sequence of labels of its edges, with the index of the leaf at the end of the sequence. Such a maximal branch is then an occurrence.

The set O_A of occurrences of a formula A is the set of maximal branches of T_A. We define a function of **linkage** $\mathcal{L}_A : O_A \to \mathbb{A} \cup \{\dagger\}$ as follows: if the leaf reached by the maximal branch a is linked to a node c, then $\mathcal{L}_A(a)$ is the sequence of labels of the edges we cross to reach c starting from the root, with a 0 at the end; otherwise, $\mathcal{L}_A(a) = \dagger$.

The structure $(O_A, \mathcal{L}_A)$ will be called a **game**. It will also be denoted A, with no risk of confusion.

Example: The type $A = \forall X_1.(X_1 \to ((\forall X_2.X_2) \to (X_3 \times \bot)))$ has as set of occurrences:

$$O_A = \{\star\downarrow 0\ ,\ \star\uparrow\downarrow\star 0\ ,\ \star\uparrow\uparrow l3\ ,\ \star\uparrow\uparrow r0\}$$

And its function of linkage is given by:

$$\begin{cases} \mathcal{L}_A(\star\downarrow 0) & = & \star 0 \\ \mathcal{L}_A(\star\uparrow\downarrow\star 0) & = & \star\uparrow\downarrow\star 0 \\ \mathcal{L}_A(\star\uparrow\uparrow l3) & = & \dagger \\ \mathcal{L}_A(\star\uparrow\uparrow r0) & = & \dagger \end{cases}$$

Definition 11 (game) *A **game** A is defined by a finite non-empty set $O_A \subseteq \mathbb{A}$ and a function of **linkage** $\mathcal{L}_A : O_A \to \mathbb{A} \cup \{\dagger\}$ satisfying the following conditions:*

- *O_A is **coherent**: for every $a \in O_A$, either $\vdash a$ or $\exists a' \in O_A,\ a' \vdash a$*
- *O_A is **non-ambiguous**: $\forall a, a' \in O_A$, if $\mathcal{E}(a) \sqsubseteq^p \mathcal{E}(a')$ then $a = a'$*
- *for every $a \in O_A$, either $\mathcal{L}_A(a) = \dagger$ or $\mathcal{L}_A(a) = a'[\star 0] \sqsubseteq^p a$ for some $a' \in \mathbb{A}$*
- *for every $a \in O_A$, if $\sharp(a) \neq 0$ then $\mathcal{L}_A(a) = \dagger$*

The set of games is denoted $\mathcal{G}$.

We stress the fact that the set O_A shall not be empty: this will be a crucial point in our proofs.

Definition 12 (auxiliary polarity) *Given a game A, we define its **auxiliary polarity** as a partial function $paux_A : O_A \rightharpoonup \{\mathbf{O}, \mathbf{P}\}$ by: $paux_A(c) = \lambda(\mathcal{L}_A(c))$ if $\mathcal{L}_A(c) \neq \dagger$, otherwise it is undefined.*

4.2 Alternative, inductive interpretation of a formula

We define the following constructions on games:
(atoms) $\bot = (\{0\}, 0 \mapsto \dagger)$ $\quad X_i = (\{i\}, i \mapsto \dagger)$ for $i > 0$.
(product) if $A, B \in \mathcal{G}$, we define $A \times B$ by:

- $\mathcal{O}_{A\times B} = \{la \mid a \in \mathcal{O}_A\} \cup \{rb \mid b \in \mathcal{O}_B\}$
- $\mathcal{L}_{A\times B}(la) = \begin{cases} \dagger & \text{if } \mathcal{L}_A(a) = \dagger \\ l\mathcal{L}_A(a) & \text{otherwise} \end{cases} \qquad \mathcal{L}_{A\times B}(rb) = \begin{cases} \dagger & \text{if } \mathcal{L}_B(b) = \dagger \\ r\mathcal{L}_B(b) & \text{otherwise} \end{cases}$

(arrow) if $A, B \in \mathcal{G}$, we define $A \to B$ by:

- $\mathcal{O}_{A\to B} = \{\downarrow a \mid a \in \mathcal{O}_A\} \cup \{\uparrow b \mid b \in \mathcal{O}_B\}$
- $\mathcal{L}_{A\to B}(\downarrow a) = \begin{cases} \dagger & \text{if } \mathcal{L}_A(a) = \dagger \\ \downarrow\mathcal{L}_A(a) & \text{otherwise} \end{cases} \qquad \mathcal{L}_{A\to B}(\uparrow b) = \begin{cases} \dagger & \text{if } \mathcal{L}_B(b) = \dagger \\ \uparrow\mathcal{L}_B(b) & \text{otherwise} \end{cases}$

(quantification) if $A \in \mathcal{G}$ and $i > 0$, we define $\forall X_i.A$ by:

- $\mathcal{O}_{\forall X_i.A} = \{\star a \mid a \in \mathcal{O}_A \wedge \sharp(a) \neq i\} \cup \{\star a[0] \mid a \in \mathcal{O}_A \wedge \sharp(a) = i\}$
- $\mathcal{L}_{\forall X_i.A}(\star a) = \begin{cases} \dagger & \text{if } \mathcal{L}_A(a) = \dagger \\ \star\mathcal{L}_A(a) & \text{otherwise} \end{cases} \qquad \mathcal{L}_{\forall X_i.A}(\star a[0]) = \star 0$

This gives rise to an inductive interpretation of a formula, which coincides with the one defined from the syntactic tree.

Finally, we define an operation of substitution on games:

Definition 13 (substitution) *Let $A, B \in \mathcal{G}$. The* ***substitution*** *of X_i by B in A is the game $A[B/X_i]$ defined by:*

- $\mathcal{O}_{A[B/X]} = \{a \in \mathcal{O}_A \mid \sharp(a) \neq i\} \cup \{a[b] \mid a \in \mathcal{O}_A \wedge \sharp(a) = i \wedge b \in \mathcal{O}_B\}$
- $\mathcal{L}_{A[B/X]}(a) = \mathcal{L}_A(a)$ *and* $\mathcal{L}_{A[B/X]}(a[b]) = \begin{cases} \dagger & \textit{if } \mathcal{L}_B(b) = \dagger \\ a[\mathcal{L}_B(b)] & \textit{otherwise} \end{cases}$

One can check that this coincides with the operation of substitution on formulas.

5 The typed model

5.1 Moves and strategies on a game

We are now going to describe how we can play in a game. We will take advantage of the way we have defined games: whereas in many second order game models like [Hug00] or [dL07] moves have a complex structure, here they will be easy to derive from $\mathcal{O}_A$ and $\mathcal{L}_A$.

As in [AJ03], the intuition is that a move in A can either be built directly from an occurrence of $\mathcal{O}_A$, or it can be decomposed as $m_1[m_2]$, where m_1 is built from an occurrence of $\mathcal{O}_A$ and m_2 is a move in another game B which substitutes a quantifier.

Note that the moves and strategies defined this way do not constitute the morphisms of our model, but they will be used as interpretations of Church-style terms.

We introduce the grammar of **typed moves**:

$$m ::= \uparrow m \mid \downarrow m \mid rm \mid lm \mid \star^B m \mid j \qquad (B \in \mathcal{G}, j \in \mathbb{N})$$

These moves form the set $\mathbb{M}$.

The operation of **anonymity** $\mathcal{A} : \mathbb{M} \to \mathbb{A}$ erases the game indication in a typed move:

- $\mathcal{A}(i) = i$ for $i \geq 0$
- $\mathcal{A}(\star^A m) = \star\mathcal{A}(m)$
- $\mathcal{A}(\alpha m) = \alpha\mathcal{A}(m)$ for $\alpha \in \{r, l, \uparrow, \downarrow\}$.

For $m \in \mathbb{M}$ and $a \in \mathbb{A}$, we define a partial operation of **formula extraction** $\frac{m}{a}$ by:

- $\frac{\star^B m}{\star 0} = B$
- if $\frac{m}{a}$ is defined, $\frac{\star^B m}{\star a} = \frac{\alpha m}{\alpha a} = \frac{m}{a}$ where $\alpha \in \{\uparrow, \downarrow, r, l\}$

Definition 14 (moves of a game) *Let A be a game. Its set of **moves** $\mathcal{M}_A \subseteq \mathbb{M}$ is given by defining the relation $m \in \mathcal{M}_A$ by induction on m:*

- *if $\mathcal{A}(m) = a \in \mathcal{O}_A$ and $\mathcal{L}_A(a) = \dagger$ then $m \in \mathcal{M}_A$*
- *if $m = m_1[m_2]$, where $\mathcal{A}(m_1) = a \in \mathcal{O}_A$, $\mathcal{L}_A(a) \neq \dagger$ and $m_2 \in \mathcal{M}_B$ with $B = \frac{m_1}{\mathcal{L}_A(a)}$, then $m \in \mathcal{M}_A$.*

This definition is well-defined, because in the second case we necessarily have at least one token $\star^B$ in m_1, so the size of m_2 is strictly smaller than the size of $m_1[m_2]$: that is why we say that the definition is inductive.

Example: Let us recall the type $A = \forall X_1.(X_1 \to ((\forall X_2.X_2) \to (X_3 \times \bot)))$ of the preceding example. One possible way to "play a move" in this game[3] is to instantiate the variable X_1 with a type B (take $B = \bot \times X_3$ for example), then to go on the left side of the first arrow and to play a move of B.

This corresponds to a move like $m = \star^B \downarrow r3$. One can check with the definition that this move indeed belongs to $\mathcal{M}_A$: $m = m_1[m_2]$ with $m_1 = \star^B \downarrow 0$ and $m_2 = r3$. $\mathcal{A}(m_1) = \star\downarrow 0 \in \mathcal{O}_A$, $\mathcal{L}_A(\star\downarrow 0) = \star 0$ and $\frac{\star^B \downarrow 0}{\star 0} = B$. Moreover, $\mathcal{A}(m_2) = r3 \in \mathcal{O}_B$ and $\mathcal{L}_B(m_2) = \dagger$ so $m_2 \in \mathcal{M}_B$ (first case of the definition). So, $m \in \mathcal{M}_B$ (second case of the definition).

Intuitively, we have the following:

- m_1 is the part of the move played in A, and $c = \mathcal{A}(m_1)$ is the corresponding occurrence
- $\mathcal{L}_a(c)$ indicates where the interesting quantifier has been instantiated
- $\frac{m_1}{\mathcal{L}_A(c)} = B$ indicates by which game it has been instantiated
- m_2 is the part of the move played in B.

Definition 15 (justified sequence, play on a game) *Let A be a game and s be a play (resp. a justified sequence) on the grammar $\mathbb{M}$. If every move of s belongs to $\mathcal{M}_A$, then we say that s is a play (resp. a justified sequence) on the game A. The set of plays on the game A is denoted $\mathcal{P}_A$.*

[3]This notion is related to the idea of **evolving game** introduced in [MO01] and reused in [dL07].

Example: Let us consider the play $s = \star^B\uparrow\uparrow l3 \cdot \star^B\downarrow r3$ with $B = \bot \times X_3$. This is of course a play *in* $A = \forall X_1.(X_1 \to (\forall X_2.X_2) \to (X_3 \times \bot))$.

What is interesting to notice is that, if for example $C = X_3 \times \bot$, then the sequence $s' = \star^C\uparrow\uparrow l3 \cdot \star^B\downarrow r3$ is not a play because it is not a justified sequence: indeed, one must have $B = C$ if we want $m_2 = \star^B\downarrow r3$ to be justified by $m_1 = \star^C\uparrow\uparrow l3$.

More generally, for any move m in a play s which contains the token $\star^B$, there is a sequence of moves $m_1, \ldots, m_n$ that also contains the token $\star^B$ at the same place, with $m_n = m$ and m_i justifies m_{i+1} for $1 \leq i < n$. If this sequence is chosen to be of maximal length, then m_1 is the minimal hereditarily justifier of m which contains the token $\star^B$: it is the first time that it appears (at the right place). We will say that B is played by $\lambda(m_1)$ at the **level** of m_1. Note that $\lambda(m_1) = paux_A(m)$.

One can formalise this definition:

Definition 16 (level) *If a move m in a play $s \in \mathcal{P}_A$ contains the token $\star^B$, then it can be written $m = m_0 \star^B [m_1]$. We say that B is played (by $\lambda(m_0)$) at the **level** of m if m_1 does not contain the token $\downarrow$.*

Typed strategies are defined as expected:

Definition 17 (strategy on a game) *Let σ be a strategy on the grammar $\mathbb{M}$, we say that σ is a strategy on A and we note $\sigma :: A$ if any play of σ belongs to $\mathcal{P}_A$. We say that σ is a typed strategy in this case.*

Strategies on games have to be understood as interpretations[4] of Church-style system F terms; they will be used in the Curry-style model because we have to express in the model the fact that a well-typed Curry-style term is the erasure of a well-typed Church-style term.

5.2 Uniformity

In [dL07], we saw that strategies defined as generally as possible were not able to capture exactly the type isomorphisms of the syntax, because they were generating too many isomorphisms in the model. That is why we introduced a notion of *uniformity*, which restrained the behaviour of strategies (in order to avoid confusion, we will call *weak uniformity* the notion of uniformity defined in [dL07]; by the way, weak uniformity plays no role in the present model).

The situation is similar here: we are not able to derive the characterisation of Curry-style type isomorphisms if the well-typed Church-style terms are interpreted by the (typed) strategies defined above. So we introduce a notion of **uniformity** on these strategies.

The intuition of this notion is the following: consider an η-long, β-normal term t of the Church-style system F, and suppose $\vdash t : \forall X.A$. The term t has the form $t = \Lambda X.t'$ with $\vdash t' : A$: so it behaves like if it was instantiating the quantifier $(\forall X)$ with a variable (X). More generally, the terms of the Church-style system F should be interpreted by strategies where, each time **O** has to play a game, he gives a variable game X_i.

But these strategies (that we will call **symbolic**) do not compose: in the Church-style syntax, this corresponds to the fact that the term $\vdash t : \forall X.A$ can be instantiated at any type B through the operation $t \mapsto t\{B\}$, and so the term t can be extended to any type $A[B/X]$. In the model, this means that the symbolic strategy interpreting t must be extensible to a more complete strategy, where **O** can play any game he wants. This extension consists in playing copycat plays between the different occurrences of the variables X (like in the syntax, the η-long β-normal form of $t\{B\}$ is generated from t through η-expansions), that is why it is called the **copycat extension**.

To sum up, a uniform strategy will be a symbolic strategy extended by copycat extension. This idea has to be related with the strategies of Dominic Hughes [Hug00] and, above all, with Murawski's notion of *good strategies* [MO01]. The notion of weak uniformity discussed above is an analogous, but less restrictive, condition: uniformity implies weak uniformity. Finally, uniformity has of course a strong connection

[4]We chose not to explicit this interpretation because we do not need it; one could also prove that we have a model of Church-style system F, but it is not an important question here.

with hyperuniformity: the two notions express analogous ideas, but hyperuniformity applies on untyped strategies, whereas uniformity is formulated in a typed context, and then requires more cautiousness.

In the following definition, $\mathcal{BV}(A)$ stands for the set of bi-views in a game A, and $m[B/j]$ (resp. $s[B/j]$) is obtained from the move m (resp. the play s) by replacing each token of the form $\star^A$ by $\star^{A[B/X_j]}$. Note that $s[B/j]$ is a play, but does not necessarily belong to any $\mathcal{M}_A$ for some A: actually, this play will only be used as an intermediate construction.

Definition 18 (copycat extension of a typed play) *Let $s = m_1 \dots m_n$ be a typed play on the game A, let $B \in \mathcal{G}$ and $j > 0$.*

*We first define the **flat extension** of s: given a sequence of initial moves $r = (r_i)_{i\in\mathbb{N}}$ in $\mathcal{M}_B$, $Fl^s_{j,B}(r)$ is the play $t[B/j]$ where t is obtained from s by replacing each sequence $m_i m_{i+1}$ such that $\sharp(m_i) = j$ and $\lambda(m_i) = \mathbf{O}$ by $m_i[r_i]m_{i+1}[r_i]$.*

*Let m_i be an $\mathbf{O}$-move of s such that $\sharp(m_i) = j$, suppose $Fl^s_{j,B}(r) = s_1 m'_i[r_i]m'_{i+1}[r_i]s_2$ with $m'_i = m_i[B/j]$ and $m'_{i+1} = m_{i+1}[B/j]$, and let $v = n_1 \dots n_p \in \mathcal{BV}(B)$. The B-**copycat extension** of s at position i along the index j (with parameters v, r) is the play $s' = CC^s_{j,B}(i, v, r)$ defined by:*

- $s' = s_1 m'_i[n_1]m'_{i+1}[n_1]s_2$ *if* $p = 1$
- $s' = s_1 m'_i[n_1]m'_{i+1}[n_1]m'_{i+1}[n_2]m'_i[n_2] \dots m'_{i+1}[n_p]m'_i[n_p]$ *if p even*
- $s' = s_1 m'_i[n_1]m'_{i+1}[n_1]m'_{i+1}[n_2]m'_i[n_2] \dots m'_i[n_p]m'_{i+1}[n_p]$ *if $p > 1$ and p odd*

Definition 19 (symbolic strategy) *A play s on the game A is said to be **symbolic** if, whenever a game is played by* **O** *it is a variable game $X_i \notin FTV(A)$. These variable games are called the **copycat variables** of the play.*

A symbolic strategy is a strategy which contains only symbolic plays.

Definition 20 (copycat extension of an innocent symbolic strategy) *The copycat extension of an innocent symbolic strategy $\bar{\sigma} : A$ is the smallest innocent strategy which contains $\bar{\sigma}$ and is stable under any copycat extension along a copycat variable.*

Definition 21 (uniform strategy) *Let σ be a strategy on the game A. σ is said to be uniform if there exists a symbolic innocent strategy $\bar{\sigma}$ on A such that σ is the copycat extension of $\bar{\sigma}$.*

Proposition 2 *If $\sigma :: A \to B$ and $\tau :: B \to C$ are two uniform strategies then $\sigma; \tau :: A \to C$ is uniform.*

The proof of this proposition can be found in appendix A.

6 The Curry-style model

We are now ready to define our model: the key ingredient will be to relate untyped strategies with typed strategies through a notion of **realization**. First we relate untyped moves with typed moves through an operation of **erasure** *erase* : $\mathbb{M} \to \mathbb{X}$ defined by:

$$erase = \mathcal{E} \circ \mathcal{A}$$

Definition 22 (realization) *Let σ be an untyped strategy and $\tilde{\sigma}$ a typed strategy on A. We say that $\tilde{\sigma}$ is a **realization** of σ on A if we have: for every $sxy \in \sigma$ and $s' \in \tilde{\sigma}$, if $s'm' \in \mathcal{P}_A$ is such that $erase(s'm') = sx$ then there exists n' such that $s'm'n' \in \tilde{\sigma}$ and $erase(s'm'n') = sxy$.*

At present we have all the ingredients to define the model:

- **objects** are games
- a **morphism** between A and B is an untyped strategy σ such that:

– σ is hyperuniform
– there exists a typed strategy $\tilde{\sigma}$ which is a realization of σ on $A \to B$
– $\tilde{\sigma}$ is uniform.

In this case we note $\sigma : A \to B$.

Let us prove that we have a model of Curry-style system F indeed.

Lemma 4 *If $\sigma : A \to B$ and $\tau : B \to C$ then $\sigma;\tau : A \to C$.*

Proof: If we note $\tilde{\sigma}$ and $\tilde{\tau}$ two realizations of σ and τ respectively, we obtain a realization of $\sigma;\tau$ on $A \to C$ by taking the composite $\tilde{\sigma};\tilde{\tau}$ *in the grammar* $\mathbb{M}$. Indeed, suppose $sxy \in \sigma;\tau$, $s' \in \tilde{\sigma};\tilde{\tau}$ with $erase(s') = s$ and $s'm' \in \mathcal{P}_{A\to C}$ with $erase(s'm') = sx$. There exist an untyped justified sequence u such that $u\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} = s_1 \in \sigma$, $u\upharpoonright_{\uparrow,\downarrow\uparrow} = s_2 \in \tau$ and $u\upharpoonright_{\uparrow,\downarrow\downarrow} = sxy$, and a typed justified sequence t such that $t\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} = t_1 \in \tilde{\sigma}$, $t\upharpoonright_{\uparrow,\downarrow\uparrow} = t_2 \in \tilde{\tau}$ and $t\upharpoonright_{\uparrow,\downarrow\downarrow} = s'$.

We note $u = u_0 x b_1 \dots b_q y$, with $b_1, \dots b_q$ of shape $\downarrow\uparrow$. Suppose for example that m' is of shape $\downarrow$. Then there exists n'_1 such that $t_1 m' n'_1 \in \tilde{\sigma}$ and $erase(t_1 m' n'_1) = s_1 x b_1$; we set $T_1 = tm'n'_1$. Then there exists n'_2 such that $t_2 n'_1 n'_2 \in \tilde{\tau}$[5] and $erase(t_2 n'_1 n'_2) = s_2 b_1 b_2$; we set $T_2 = tm'n'_1n'_2$, etc. So, we construct step by step a justified sequence T such that $T\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \tilde{\sigma}$, $T\upharpoonright_{\uparrow,\downarrow\uparrow} \in \tilde{\tau}$ and $erase(T) = u$. This gives us also that $T\upharpoonright_{\uparrow,\downarrow\downarrow} = s'm'n'$ is a play, so it belongs to $\tilde{\sigma};\tilde{\tau}$ and $erase(s'm'n') = sxy$.

Finally: $\tilde{\sigma}$ and $\tilde{\tau}$ are innocent and uniform, so $\tilde{\sigma};\tilde{\tau}$ is innocent and uniform by prop. 2; σ and τ are hyperuniform so $\sigma;\tau$ is hyperuniform by lemma 2. □

Lemma 5 *If $\sigma : \Gamma \to A$ and $X_j \notin \Gamma$ then $\sigma : \Gamma \to \forall X_j.A$*

Proof: Let us consider $\tilde{\sigma} :: \Gamma \to A$ a realization of σ on $\Gamma \to A$: if $\tilde{\sigma}$ is the copycat extension of a symbolic strategy $\bar{\sigma}$, then we define the strategy $\bar{\sigma}'$ as the strategy $\bar{\sigma}$ where each move written $\uparrow m$ in a play has been replaced by $\uparrow \star^{X_j} m$. This strategy is symbolic on $\Gamma \to \forall X_j.A$, and its copycat extension $\tilde{\sigma}'$ is a realization of σ because of hyperuniformity (indeed, the only difference between $\tilde{\sigma}$ and $\tilde{\sigma}'$ is a copycat extension along X_j). □

Lemma 6 *If $\sigma : \Gamma \to \forall X_j.A$ and B is a game then $\sigma : \Gamma \to A[B/X_j]$.*

Proof: If $\tilde{\sigma}$ is a realization of σ on $\Gamma \to \forall X_j.A$, a realization $\tilde{\sigma}'$ on $\Gamma \to A[B/X_j]$ is obtained by taking only plays where each initial move takes the form $\uparrow \star^B m$, and by replacing each move $\uparrow \star^B m$ by $\uparrow m$.

Let us now prove the uniformity of $\tilde{\sigma}'$: if $\tilde{\sigma}$ is the copycat extension of a symbolic strategy $\bar{\sigma}$, we consider a view s of $\bar{\sigma}$. Let X_j be the first copycat variable appearing in s, we choose a variable $X_k \notin FTV(A) \cup FTV(B)$ and we call s_k the (unique) X_k-copycat extension of s along j. Let us define $E(s)$ as the smallest set of plays containing s_k and stable by B-copycat extensions along k. The strategy $\bar{\sigma}'$ will be the smallest innocent strategy containing all the sets $E(s)$, for s describing all the views of $\bar{\sigma}$. Then one can check that $\tilde{\sigma}'$ is the copycat extension of $\bar{\sigma}'$. □

Lemma 7 *The following holds:*

- $id : A \to A$
- $\pi_r : \Gamma \times A \to A$
- *If $\sigma : \Gamma \to A$ and $\tau : \Gamma \to B$ then $\langle \sigma, \tau \rangle : \Gamma \to (A \times B)$.*
- $eval : (A \to B) \times A \to B$

[5]More precisely $n'_1 = \uparrow n''$ should be renamed as $\downarrow n''$.

- *If $\sigma : \Gamma \times A \to B$ then $\Lambda(\sigma) : \Gamma \to (A \to B)$.*

These cases are trivial: for example, a realization of *id* on $A \to A$ is

$$\rho = \{s \in \mathcal{P}_{A\to A} \mid s \text{ of arrow shape and } \forall t \in \mathbb{E}, t \le s \Rightarrow t\upharpoonright_{\uparrow} = t\upharpoonright_{\downarrow}\}$$

and it is uniform, with symbolic strategy $\bar\rho$ defined by:

$$\bar\rho = \{s \in \mathcal{P}_{A\to A} \mid s \text{ of arrow shape, } s \text{ symbolic and } \forall t \in \mathbb{E}, t \le s \Rightarrow t\upharpoonright_{\uparrow} = t\upharpoonright_{\downarrow}\}$$

If Γ is a typing context of the form $\Gamma = x_1 : A_1, x_2 : A_2, \dots, x_n : A_n$, we define the sequence of variables $\overline{\Gamma} = x_1, x_2, \dots, x_n$ and the type $|\Gamma| = A_1 \times A_2 \times \dots \times A_n$, and we have:

Proposition 3 *If $\Gamma \vdash t : A$ then $[\![\overline{\Gamma} \vdash t]\!] : |\Gamma| \to A$.*

This, together with prop. 1, means that we have obtained a model of Curry-style system F.

7 Hyperforests

In this section we introduce the notion of **hyperforest**, an arborescent structure built from games. In [dL07], following [Hug00], we interpreted second-order types directly as hyperforests (that we called polymorphic arenas). But the substitution was difficult to define in this context, and moves had a complicated formulation; that is why in this paper we introduce hyperforests only as an indirect interpretation of types.

Hyperforests will be the fundamental structure for our work on isomorphisms.

7.1 Forests and hyperforests

In what follows, the set of subsets of a set E will be denoted $\mathbb{P}(E)$.

Definition 23 (forest) *A **forest** is an ordered set $(E, \le)$ such that, for every y in E, $\{x \mid x \le y\}$ is finite and totally ordered by $\le$. The forest is **finite** if E is finite.*

Definition 24 (hyperforest) *An **hyperforest** $H = (\mathcal{F}, \mathcal{R}, \mathcal{D})$ is a finite forest $\mathcal{F}$ together with a set of **hyperedges** $\mathcal{R} \subseteq \mathcal{F} \times \mathbb{P}(\mathcal{F})$ and a partial function of **decoration** $\mathcal{D} : \mathcal{F} \rightharpoonup \mathcal{X}$, where:*

- *for every $(t, S) \in \mathcal{R}$, if $s \in S$ then $t \le s$ and $\mathcal{D}(s)$ is undefined*
- *for every $b = (t, S)$ and $b' = (t', S')$ in $\mathcal{R}$, $S \cap S' \neq \emptyset \Rightarrow b = b'$*

We note $\mathcal{T}^H = \{t \in \mathcal{F} \mid \exists S \subseteq \mathcal{F},\ (t, S) \in \mathcal{R}\}$ and $\mathcal{S}^H = \{s \in \mathcal{F} \mid \exists (t, S) \in \mathcal{R},\ s \in S\}$.

Definition 25 (reference, friends) *Let $H = (\mathcal{F}, \mathcal{R}, \mathcal{D})$ be an hyperforest. For any $s \in \mathcal{F}$, if $s \in \mathcal{S}^H$ then there exists $(t, S) \in \mathcal{R}$ with $s \in S$: the **reference** of s is defined as $ref^H(s) = t$ and the set of **friends** of s is $fr^H(s) = S \backslash \{s\}$. If $s \notin \mathcal{S}^H$, ref^H and fr^H are not defined in s.*

We are now going to exhibit the hyperforest structure associated with a game A.

7.2 From partially ordered sets to forests

Let $(E, \le)$ be a partially ordered set. The relation $\vdash \subseteq E \cup (E \times E)$ is given by:

$$\begin{cases} \vdash e & \text{iff } e' \le e \Rightarrow (e' = e) \\ e \vdash e' & \text{iff } e \le e' \wedge \forall f,\ e \le f \le e' \Rightarrow (e = f \vee e' = f) \end{cases}$$

One defines the set F of **paths** in $(E, \le)$, i.e. the set of sequences $e_1 e_2 \dots e_n$ of elements of E such that $\vdash e_1$ and $e_i \vdash e_{i+1}$ for $1 \le i \le n-1$. If we consider the prefix ordering $\le'$ on F, then $(F, \le')$ is a forest.

We also define the operation $or : F \to E$ by $or(f) = e_n$ if $f = e_1 \dots e_n$ ($or(f)$ is called the **origin** of f).

7.3 From games to hyperforests

If A is a game, $\mathcal{O}_A$ is a finite partially ordered set, to which one can associate a forest $\mathcal{F}_A$ through the preceding construction. Extending $\vdash$ to $\mathcal{F}_A$ generates the enabling relation of the forest: this justifies *a posteriori* the definition of an enabling relation for arbitrary moves given in section 2.

Furthermore, one deduces from $\mathcal{L}_A$ the relation $\mathcal{R}_A \subseteq \mathcal{F}_A \times \mathbb{P}(\mathcal{F}_A)$ as follows: let $\mathcal{L} = \{a[\star 0] \in \mathbb{A} \mid \exists a' \in \mathcal{O}_A, a[\star 0] \sqsubseteq^p a'\}$. Then :

$$(t, S) \in \mathcal{R}_A \text{ iff there exists } y \in \mathcal{L} \text{ such that, for every } s \in S\text{:}$$

- $\mathcal{L}_A(or(s)) = y$
- $t \leq s$
- $y \sqsubseteq^p or(t)$
- for every $t' \leq t$, $y \sqsubseteq^p or(t')$ implies $t' = t$.

One also defines the partial function $\mathcal{D}_A : \mathcal{F}_A \rightharpoonup \mathcal{X}$ by: $\mathcal{D}_A(x) = X_i$ iff $\sharp(or(x)) = i$ $(i > 0)$.

Then we have:

Lemma 8 *If A is a game, then $H_A = (\mathcal{F}_A, \mathcal{R}_A, \mathcal{D}_A)$ is an hyperforest.*

Example: Consider the type $A = \forall X_1.((X_1 \times X_2) \to (X_1 \times \bot))$. We have:

$$\mathcal{O}_A = \{\star\downarrow l0, \star\downarrow r2, \star\uparrow l0, \star\uparrow r0\}$$

and:

$$\begin{cases} \mathcal{L}_A(\star\downarrow l0) &=& \star 0 \\ \mathcal{L}_A(\star\downarrow r2) &=& \dagger \\ \mathcal{L}_A(\star\uparrow l0) &=& \star 0 \\ \mathcal{L}_A(\star\uparrow r0) &=& \dagger \end{cases}$$

The paths are: $a = \star\uparrow l0$, $b = \star\uparrow l0 \cdot \star\downarrow l0$, $c = \star\uparrow l0 \cdot \star\downarrow r2$, $d = \star\uparrow r0$, $e = \star\uparrow r0 \cdot \star\downarrow l0$ and $f = \star\uparrow r0 \cdot \star\downarrow r2$. Besides, $\mathcal{L} = \{\star 0\}$.

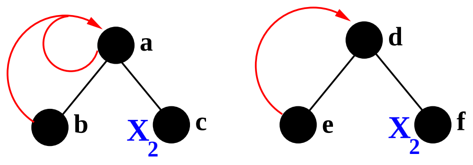
Hence the hyperforest H_A is given by:

$$\mathcal{F}_A = \{a, b, c, d, e, f\}$$

$$\mathcal{R}_A = \{(a, \{a, b\}), (d, \{e\})\}$$

$$\mathcal{D}_A(c) = \mathcal{D}_A(f) = X_2$$

This can be resume in the following representation of H_A:



One can extend the definition of polarity to the nodes of the hyperforest: if A is a game with associated hyperforest $H_A = (\mathcal{F}_A, \mathcal{R}_A, \mathcal{D}_A)$, then for $a \in \mathcal{F}_A$ we define $\lambda(a) = \lambda(or(a))$. This coincides with an alternative definition of polarity, which is common in arena games: $\lambda(a) = \mathbf{O}$ (resp. $\lambda(a) = \mathbf{P}$) if the set $\{a' \in \mathcal{F}_A \mid a' \leq a\}$ has an odd cardinality (resp. an even cardinality). Note also that $paux_A(or(a)) = \lambda(ref_A(a))$.

Finally, if A is a game, we note:

$$fr_A = fr^{H_A} \qquad ref_A = ref^{H_A} \qquad \mathcal{S}_A = \mathcal{S}^{H_A} \qquad \mathcal{T}_A = \mathcal{T}^{H_A}$$

Note that the nodes of the forest $\mathcal{F}_A$ contain "more information" than the occurrences of $\mathcal{O}_A$. Indeed, given a node $c \in \mathcal{F}_A$, one is able to give the ordered list of its ancestors, whereas for an occurrence we may have many ancestors that are not compatible one with the order for the ordering. This idea will be used in the proof of theorem 1 to reason about plays with nodes instead of occurrences.

8 Type isomorphisms

8.1 Isomorphisms in the model

Definition 26 (Church-isomorphism) *Let $H_1 = (\mathcal{F}_1, \mathcal{R}_1, \mathcal{D}_1)$ and $H_2 = (\mathcal{F}_2, \mathcal{R}_2, \mathcal{D}_2)$ be two hyperforests. We say that H_1 and H_2 are **Church-isomorphic** ($H_1 \simeq_{Ch} H_2$) if there exists a bijection $f : \mathcal{F}_1 \to \mathcal{F}_2$ which preserves the hyperforest structure, i.e. such that:*

- $a \leq a'$ *iff* $f(a) \leq f(a')$
- $\mathcal{R}_2 = f(\mathcal{R}_1)$
- $\mathcal{D}_2 \circ f = \mathcal{D}_1$

Definition 27 (Curry-isomorphism) *Let $H_1 = (\mathcal{F}_1, \mathcal{R}_1, \mathcal{D}_1)$ and $H_2 = (\mathcal{F}_2, \mathcal{R}_2, \mathcal{D}_2)$ be two hyperforests. We say that H_1 and H_2 are **Curry-isomorphic** ($H_1 \simeq_{Cu} H_2$) if there exists a bijection $f : \mathcal{F}_1 \to \mathcal{F}_2$ such that:*

- $a \leq a'$ *iff* $f(a) \leq f(a')$
- $\mathcal{S}^{H_2} = f(\mathcal{S}^{H_1})$
- *for every $(t, S) \in \mathcal{R}_1$ (resp. $(t, S) \in \mathcal{R}_2$), if there exists $s \in S$ such that $\lambda(s) \neq \lambda(t)$, then $(f(t), f(S)) \in \mathcal{R}_2$ (resp. $(f^{-1}(t), f^{-1}(S)) \in \mathcal{R}_1$)*
- $\mathcal{D}_2 \circ f = \mathcal{D}_1$.

Definition 28 (game isomorphism) *A **game isomorphism** between two games A and B is a couple of untyped strategies $\sigma : A \to B$ and $\tau : B \to A$ such that $\sigma;\tau = \tau;\sigma = id$. We note $A \simeq_g B$ if there is a game isomorphism between A and B.*

We are now able to formulate the key theorem of our paper. This theorem provides a geometrical characterisation of isomorphisms in the model, which is the core of the proof of equational characterisation for the syntax.

Theorem 1 *Let $A, B \in \mathcal{G}$. If there exists a game isomorphism (σ, τ) between A and B ($A \simeq_g B$) then their hyperforests are Curry-isomorphic ($H_A \simeq_{Cu} H_B$).*

The proof of this theorem can be found in appendix B.

8.2 Characterisation of Curry-style type isomorphisms

Proving theorem 1 was the main step towards the characterisation of Curry-style isomorphisms: we are now able to establish our final result.

Let us recall the equational system $\simeq_\varepsilon$ which we claim to characterise Curry-style type isomorphisms:

$$\begin{array}{rcll}
A \times B & \simeq_\varepsilon & B \times A & \\
A \times (B \times C) & \simeq_\varepsilon & (A \times B) \times C & \\
A \to (B \to C) & \simeq_\varepsilon & (A \times B) \to C & \\
A \to (B \times C) & \simeq_\varepsilon & (A \to B) \times (A \to C) & \\
\forall X.\forall Y.A & \simeq_\varepsilon & \forall Y.\forall X.A & \\
A \to \forall X.B & \simeq_\varepsilon & \forall X.(A \to B) & \text{if } X \notin FTV(A) \\
\forall X.(A \times B) & \simeq_\varepsilon & \forall X.A \times \forall X.B & \\
\forall X.A & \simeq_\varepsilon & A[\forall Y.Y/X] & \text{if } X \notin Neg_A
\end{array}$$

Lemma 9 *Let A and B be two types such that the hyperforests H_A and H_B are Curry-isomorphic. Then A and B are equal up to the equational system $\simeq_\varepsilon$.*

Proof: Let A' and B' be the normal forms of A and B for the following rewriting system:

$$\forall X.C \Rightarrow C[\forall Y.Y/X] \quad \text{if } X \notin Neg_C \text{ and } C \neq X$$

If $D_1 = \forall X.C$ and $D_2 = C[\forall Y.Y/X]$ with $X \notin Neg_C$, then $H_{D_1} \simeq_{Cu} H_{D_2}$: indeed, the bijection $f : \mathcal{F}_{D_1} \to \mathcal{F}_{D_2}$ which preserves the ordering and such that $\mathcal{S}_{D_2} = f(\mathcal{S}_{D_1})$ and $\mathcal{D}_{D_2} \circ f = \mathcal{D}_1$ is easy to define (in fact $\mathcal{O}_{D_1}$ and $\mathcal{O}_{D_2}$ are already in bijection). The fact that $X \notin Neg_C$ precisely implies that, for any $(t, S) \in \mathcal{R}_{D_1}$ corresponding to the quantification $\forall X$ (i.e. such that $\mathcal{L}_{\forall X.A}(or(s)) = \star 0$ for every $s \in S$), there is no $s \in S$ such that $\lambda(s) \neq \lambda(t)$. Reciprocally, if for any $(t, S) \in \mathcal{R}_{D_2}$ corresponding to a quantification $\forall Y.Y$, $S = \{t\}$ so there is no $s \in S$ such that $\lambda(s) \neq \lambda(t)$. Any other hyperedge is preserved by f.

Moreover, being Curry-isomorphic is a congruence (i.e. it is preserved by context), so $H_A \simeq_{Cu} H_{A'}$, $H_B \simeq_{Cu} H_{B'}$, and hence $H_{A'} \simeq_{Cu} H_{B'}$. $H_{A'}$ and $H_{B'}$ are such that for every $(t, S) \in \mathcal{R}_{A'}$ (or $(t, S) \in \mathcal{R}_{B'}$), either $S = \{t\}$ or S contains a node s with $\lambda(t) \neq \lambda(s)$. Because of the definitions of $\simeq_{Cu}$ and $\simeq_{Ch}$, this implies $H_{A'} \simeq_{Ch} H_{B'}$.

It has already been proved in [dL07][6] that in this case $A' \simeq'_\varepsilon B'$, where $\simeq'_\varepsilon$ is the same equational system as $\simeq_\varepsilon$, except that it does not make use of the last equation. Hence, we have $A \simeq_\varepsilon B$. □

Theorem 2 *Two types A and B are isomorphic in Curry-style system F if and only if $A \simeq_\varepsilon B$.*

Proof: The implication comes from the fact that we have a model (so, each type isomorphism in Curry-style system F implies a game isomorphism) and from theorem 1 and lemma 9.

For the reciprocal, we already know from [DC95] the existence in the Church-style system F of the isomorphisms corresponding to each equation of $\simeq_\varepsilon$, except the last one ($\forall X.A \simeq_\varepsilon A[\forall Y.Y/X]$ if $X \notin Neg_A$). This implies their existence in the Curry-style system F.

Hence, we need, given a type A such that $X \notin Neg_A$, to find two Curry-style terms $t : \forall X.A \to A[\forall Y.Y/X]$ and $u : A[\forall Y.Y/X] \to \forall X.A$ which compose in both ways to give the identity. We suppose Y does not appear at all in A, even as a bounded variable.

We take $t = \lambda x.x$: indeed, the identity can be shown to be of type $\forall X.A \to A[\forall Y.Y/X]$ through the following type derivation:

$$\dfrac{\dfrac{x : \forall X.A \vdash x : \forall X.A}{x : \forall X.A \vdash x : A[\forall Y.Y/X]}}{\vdash \lambda x.x : \forall X.A \to A[\forall Y.Y/X]}$$

[6]In [dL07] the interpretation of types was directly hyperforests.

t is easy to build: consider the Church-style term $M = \lambda x^{\forall X.A}.(x)\{\forall Y.Y\}$. We have $\vdash M : \forall X.A \to A[\forall Y.Y/X]$ in Church-style system F, and t is the λ-term obtained by erasing each type indication in M. Then we necessarily have $\vdash t : \forall X.A \to A[\forall Y.Y/X]$, and besides $t = \lambda x.x$.

To define u, let us consider the Church-style term P which is the η-long normal form of the identity on $A[\forall Y.Y/X]$. This term takes the form $P = \lambda x^{A[\forall Y.Y/X]}.P'$. Now consider the Church-style term Q obtained from P' by replacing each occurrence of $y\{Z\}$, where Z is some type variable and y has the type $\forall Y.Y$ coming from the substitution of X, by $y\{X\}$. For example, if $A = X \to \bot \to \bot$, this would give us $Q = (x)\lambda y^{(\forall Y.Y)\to\bot}.(y)\lambda z^{\forall Y.Y}.(z)\{X\}$

Then we introduce the Church-style term $N = \lambda x^{A[\forall Y.Y/X]}.\Lambda X.Q$, and we can check that $\vdash N : A[\forall Y.Y/X] \to \forall X.A$ in Church-style system F. u is now defined to be the erasure of N. Then we necessarily have $\vdash u : A[\forall Y.Y/X] \to \forall X.A$, and besides $u = \lambda x.x$ (modulo η-reductions) because we only modified the type indications when going from P to N.

Finally, t and u trivially compose to give the identity in both directions. $\square$

Conclusion

We have proved that type isomorphisms in Curry-style system F can be characterised by adding to the equational system of Church-style system F isomorphisms a new, non-trivial equation: $\forall X.A \simeq_\varepsilon A[\forall Y.Y/X]$ if $X \notin Neg_A$. Otherwise said, this equation characterises all the new type equivalences one can generate by erasing type indications in Church-style terms.

We used a game semantics model in order to take advantage of its dynamical and geometrical properties. The main features of the model were however often inspired by a precise analysis of the syntax: indeed, an interpretation of the quantifier as an intersection (or a lower bound like in [Chr03]) was not precise enough to be able to characterise type isomorphisms.

One can notice that our type system does not contain the type $\top$; correspondingly, our model has no empty game. This is because the rule generally associated to $\top$ takes the form: $t = \star$ if $\Gamma \vdash t : \top$. This rule is of course difficult to insert in a Curry-style setting, where terms are not typed a priori, and we have no clue whether such a rule can be adapted to this context. Anyway, the introduction of an empty game in the model would break the proof and, more interestingly, give raise to new isomorphisms like $\forall X.(X \to \bot) \simeq_g \bot$. The characterisation of isomorphisms in this model, and the possible connection with an actual syntax, have to be explored.

But the main trail of future exploration concerns parametric polymorphism. The notion of relational parametricity, introduced by Reynolds [Rey83], comes historically from the idea that a second-order function shall not depend on the type at which it is instantiated. This has led first to a semantic definition of parametricity, then to a syntactic formalisation of this notion, first by Abadi-Cardelli-Curien [ACC93] and then by Plotkin-Abadi [PA93]. Dunphy [Dun02] recently gave a categorical characterisation of parametric polymorphism.

The great advantage of parametric models is that second-order enjoys nice and natural properties in these models. For example:

- $\forall X.X \to X$ is a terminal object
- $\forall X.(A \to B \to X) \to X$ is a product of A and B
- $\forall X.X$ is an initial object
- $\forall X.(A \to X) \to (B \to X) \to X$ is a coproduct of A and B.

All these properties are of course wrong in the model described in the present paper.

Trying to build a parametric game model is a highly appealing challenge: one would be glad to extend the concrete notions and flexible features of games into a context where parametricity is understood. Studying isomorphisms in this context would be a natural question, considering the particularly powerful ones corresponding to the above properties.

Finally, relational parametricity seems to be related to Curry-style system F, if we believe in a conjecture of Abadi-Cardelli-Curien which says the following: suppose you have two terms of type A whose type erasures are the same. Then they are parametrically equal (the converse is false). This means that the parametric equality is (strictly) stronger than the Curry-style equality: the study on both Curry-style system F and parametricity in the context of games may help to explore this question.

References

[ACC93] Martin Abadi, Luca Cardelli, and Pierre-Louis Curien. Formal parametric polymorphism. *Theoretical Computer Science*, 121:9–58, 1993.

[AJ03] Samson Abramsky and Radha Jagadeesan. A game semantics for generic polymorphism. In Andrew D. Gordon, editor, *Foundations of Software Science and Computational Structures*, volume 2620 of *LNCS*, pages 1–22. Springer, 2003.

[AJM00] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF. *Information and Computation*, 163(2):409–470, December 2000.

[AM99] Samson Abramsky and Guy McCusker. Full abstraction for idealized algol with passive expressions. *Theoretical Computer Science*, 227:3–42, September 1999.

[BP01] Gilles Barthe and Olivier Pons. Type isomorphisms and proof reuse in dependent type theory. In F. Honsell and M. Miculan, editors, *Foundations of Software Science and Computation Structures*, volume 2030 of *LNCS*, 2001.

[Chr03] Julius Chroboczek. *Game Semantics and Subtyping*. Ph.D. thesis, University of Edinburgh, 2003.

[DC95] Roberto Di Cosmo. *Isomorphisms of Types*. Progress in Theoretical Computer Science. Birkhäuser, 1995.

[dL07] Joachim de Lataillade. Second-order type isomorphisms through game semantics. To appear in *Annals of Pure and Applied Logic*, Special Issue on Game Semantics, 2007. Available at `http://www.pps.jussieu.fr/~delatail/isotypes.pdf`.

[Dun02] Brian Patrick Dunphy. *Parametricity as a notion of uniformity in reflexive graphs*. Ph.D. thesis, University of Illinois, 2002.

[HO00] Martin Hyland and Luke Ong. On full abstraction for PCF. *Information and Computation*, 163(2):285–408, December 2000.

[Hug00] Dominic Hughes. *Hypergame semantics: full completeness for system F*. D.Phil. thesis, Oxford University, 2000.

[Lai97] James Laird. Full abstraction for functional languages with control. In *Proceedings of the twelfth annual symposium on Logic In Computer Science*, pages 58–67, Warsaw, June 1997. IEEE, IEEE Computer Society Press.

[Lau05] Olivier Laurent. Classical isomorphisms of types. *Mathematical Structures in Computer Science*, 15(5):969–1004, October 2005.

[MO01] Andrzej Murawski and Luke Ong. Evolving games and essential nets for affine polymorphism. In Samson Abramsky, editor, *Typed Lambda Calculi and Applications '01*, volume 2044 of *LNCS*. Springer, 2001.

[PA93] Gordon Plotkin and Martín Abadi. A logic for parametric polymorphism. In M. Bezem and J. F. Groote, editors, *International Conference on Typed Lambda Calculi and Applications*, pages 361–375, Utrecht, The Netherlands, 1993. Springer-Verlag.

[Rey83] John C. Reynolds. Types, abstraction and parametric polymorphism. In *International Federation for Information Processing Congress*, pages 513–523, 1983.

[Rit91] Mikael Rittri. Using types as search keys in function libraries. *Journal of Functional Programming*, 1(1):71–89, 1991.

A Uniform strategies compose

Proposition 2 *If $\sigma :: A \to B$ and $\tau :: B \to C$ are two uniform strategies then $\sigma;\tau :: A \to C$ is uniform.*

Proof: Consider the following strategy

$$\bar{\rho} = \{u\upharpoonright_{\uparrow,\downarrow\downarrow} \mid u \in \mathbf{Int} \wedge u\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \sigma \wedge u\upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau \wedge u\upharpoonright_{\uparrow,\downarrow\downarrow} \text{ symbolic play}\}$$

It is an innocent strategy on $A \to C$ (the proof is the same as in HON models), and it is of course symbolic. We call ρ its copycat extension, and we want to prove that $\rho = \sigma;\tau$.

First we prove that $\rho \subseteq \sigma;\tau$: as $\bar{\rho} \subseteq \sigma;\tau$, we need to show that $\sigma;\tau$ is stable by any copycat extension along any index j. Note that, if the variable game X_j is played by **O** in $A \to C$, it is also played by **O** in $A \to B$ or $B \to C$. Consider the play $s' = Fl^{s}_{j,D}(r)$ for $s = m_1 \dots m_n \in \sigma;\tau$, $D \in \mathcal{G}$ and r sequence of initial move in $\mathcal{M}_D$. One shows that $s' \in \sigma;\tau$: indeed there exist a justified sequence u and two plays $s_1 \in \sigma$ and $s_2 \in \tau$ such that $u\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} = s_1$, $u\upharpoonright_{\uparrow,\downarrow\uparrow} = s_2$ and $u\upharpoonright_{\uparrow,\downarrow\downarrow} = s$. Let us consider the justified sequence U_0 obtained from u by replacing each sequence $m_i b_1 \dots b_q m_{i+1}$ by $m_i[r_i]b_1[r_i]\dots b_q[r_i]m_{i+1}[r_i]$, and set $U = U_0[D/j]$. Then $U\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} = s'_1 \in \sigma$ (it is a flat extension, hence a copycat extension, of s_1), $U\upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau$ (it is a flat extension, hence a copycat extension, of s_2) and $U\upharpoonright_{\uparrow,\downarrow\downarrow} = s' \in \sigma;\tau$.

Now consider a move m_i of s such that $\sharp(m_i) = j$ and a bi-view $v = n_1 \dots n_p$ in the game D, and set $S = CC^{s}_{j,D}(i,v,r)$. If $U = U_1 m'_i[r_i]b_1[r_i]\dots b_q[r_i]m'_{i+1}[r_i]U_2$ with $m'_i = m_i[D/j]$ and $m'_{i+1} = m_{i+1}[D/j]$, one can build another justified sequence U', depending on the value of p:

- if $p = 1$, $U' = U_1 m'_i[n_1]b_1[n_1]\dots b_q[n_1]m'_{i+1}[n_1]U_2$
- if p even,
  $U' = U_1 m'_i[n_1]b_1[n_1]\dots b_q[n_1]m'_{i+1}[n_1]m'_{i+1}[n_2]b_q[n_2]\dots b_1[n_2]m'_i[n_2]\dots\dots m'_{i+1}[n_p]b_q[n_p]\dots b_1[n_p]m'_i[n_p]$
- if p odd and $p > 1$,
  $U' = U_1 m'_i[n_1]b_1[n_1]\dots b_q[n_1]m'_{i+1}[n_1]m'_{i+1}[n_2]b_q[n_2]\dots b_1[n_2]m'_i[n_2]\dots\dots m'_i[n_p]b_1[n_p]\dots b_q[n_p]m'_{i+1}[n_p]$

$U'\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}$ is a copycat extension of s_1 (s'_1 was the flat extension) so $U'\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \sigma$, and similarly $U'\upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau$. $U'\upharpoonright_{\uparrow,\downarrow\downarrow}$ is a play so $U'\upharpoonright_{\uparrow,\downarrow\downarrow} = S \in \sigma;\tau$.

The last thing to prove is that $\sigma;\tau \subseteq \rho$. We suppose that σ and τ are the copycat extensions of the symbolic strategies $\bar{\sigma}$ and $\bar{\tau}$ respectively. Consider a play $s \in \sigma;\tau$, there exists a justified sequence u for which $u\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} = s_1 \in \sigma$, $u\upharpoonright_{\uparrow,\downarrow\uparrow} = s_2 \in \tau$ and $u\upharpoonright_{\uparrow,\downarrow\downarrow} = s$.

Let $D_1,\dots,D_N$ be the sequence of games played by **O** in u at the level of moves of shape $\uparrow$ or $\downarrow\downarrow$. Suppose for simplicity that $X_1,\dots,X_N \notin FTV(A)$. Consider a subsequence $U = m[m_1]b_1[m_2]\dots b_q[m_{q+1}]n[m_{q+2}]$ of u such that: $\mathcal{A}(m),\mathcal{A}(b_1),\dots,\mathcal{A}(b_q),\mathcal{A}(n) \in \mathcal{O}_{(A\to B)\to C}$, $paux_{(A\to B)\to C}(c) = \mathbf{O}$ if $c = \mathcal{A}(m)$, and $b_1,\dots,b_q$ are of shape $\downarrow\uparrow$ whereas m,n are not of this shape. Suppose U is the first such sequence in u and m is of shape $\uparrow$ (the case where m is of shape $\downarrow\downarrow$ is similar). Then $\frac{m}{\mathcal{L}_{(A\to B)\to C}(c)} = D_j$ for some $1 \leq j \leq N$ (it is a game played by **O**, because $paux_{(A\to B)\to C}(c) = \mathbf{O}$).

If $u = u_1 U u_2$, we build a new sequence u' as follows:

- $t_1 = (u_1 m[m_1]b_1[m_2])\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \sigma$ is a D_j-copycat extension of some $\bar{s}_1 \in \bar{\sigma}$: indeed, σ is the smallest innocent strategy that contains **O** and is stable by copycat extension, so t_1 must be composed of many views that are obtained from $\bar{\sigma}$ by copycat extensions; besides, U is the first subsequence of its kind, so there is in fact only one of these copycat extensions that applies on a variable played at the level of a move of shape $\uparrow$ or $\downarrow\downarrow$ (so, only one B_j-extension). $\bar{s}_1$ takes the form $\bar{s}_1 = (u'_1 m[j]b_1[M_1])\upharpoonright_{\downarrow\uparrow,\downarrow\downarrow}$ where u'_1 is obtained by replacing each occurrence of D_j in u_1 by X_j
- $t_2 = (u_1 b_1[m_2]b_2[m_3])\upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau$ is a D_j-copycat extension of some $\bar{s}_2 \in \bar{\tau}$: indeed, τ is the smallest innocent strategy that contains **O** and is stable by copycat extension, so t_2 must be composed of many views that are obtained from $\bar{\tau}$ by copycat extensions; besides, U is the first subsequence of its kind, so there is in fact only one of these copycat extensions that applies on a variable played at the level of a move of shape $\uparrow$ or $\downarrow\downarrow$ (so, only one B_j-extension). $\bar{s}_2$ takes the form $\bar{s}_2 = (u'_1 m[j]b_1[M_1])\upharpoonright_{\uparrow,\downarrow\uparrow}$

- we iterate this process until we get to $n[M_{q+1}]$ for some M_{q+1}: this gives us a justified sequence u' on $(A \to B) \to C$ which can be copycat extended to $u_1 U$, and such that $u' \upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \sigma$, $u' \upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau$.

Now we iterate this process for each subsequence of u having the same properties as U, and what we obtain is a justified sequence u'' on $(A \to B) \to C$ such that $u' \upharpoonright_{\downarrow\uparrow,\downarrow\downarrow} \in \sigma$, $u' \upharpoonright_{\uparrow,\downarrow\uparrow} \in \tau$ and $t = u'' \upharpoonright_{\uparrow,\downarrow\downarrow}$ is a play. Moreover each D_j has been replaced by X_j (it might actually not be the case if some D_j did not correspond to any of our sequences, but in this case we just replace D_j by X_j harmlessly), so $t \in \bar{\rho}$.

Finally, u can be obtained from u'' by copycat extension, so s can be obtained from t by copycat extension. Hence, $s \in \rho$. □

B Proof of $A \simeq_g B \Rightarrow H_A \simeq_{Cu} H_B$

Definition 29 (zig-zag play) *A play s of arrow shape is said to be **zig-zag** if*

- *each Player move following an Opponent move of the form $\uparrow m$ (resp. $\downarrow m$) has the form $\downarrow m'$ (resp. $\uparrow m'$)*
- *each (Player) move which follows an (Opponent) initial move is justified by it*
- *$s \upharpoonright_{\uparrow}$ and $s \upharpoonright_{\downarrow}$ have the same pointers.*

If s is a zig-zag even-length play, we note $\check{s}$ the unique zig-zag play such that $\check{s} \upharpoonright_{\uparrow} = s \upharpoonright_{\downarrow}$ and $\check{s} \upharpoonright_{\downarrow} = s \upharpoonright_{\uparrow}$.

Theorem 1 *Let $A, B \in \mathcal{G}$. If there exists a game isomorphism (σ, τ) between A and B ($A \simeq_g B$) then their hyperforests are Curry-isomorphic ($H_A \simeq_{Cu} H_B$).*

Proof: For the sake of simplicity, we will throughout this proof identify the nodes of $\mathcal{F}_A$ (resp. of $\mathcal{F}_B$) with the corresponding nodes of $\mathcal{F}_{A \to B}$.

Zig-zag property

Let $\sigma : A \to B$ and $\tau : B \to A$ be the untyped strategies which form the game isomorphism, and let $\tilde{\sigma} :: A \to B$ and $\tilde{\tau} :: B \to A$ be two realizations of σ and τ, respectively on $A \to B$ and $B \to A$.

We begin with the following:

- every play of σ or τ is zig-zag
- $\tau = \{\check{s} \mid s \in \sigma\}$
- σ and τ are total on the shape $\{\uparrow, \downarrow\}$.

This has been proven in a simply typed context, i.e. with strategies playing on forests, in [Lau05]. The present situation is actually a particular case of the simply typed one where the two forests to consider are universal (in the sense that they contain any move). Totality for universal forests immediately implies totality on the arrow shape.

One consequence of totality on the arrow shape is that, whenever $s \in \tilde{\sigma}$, we have $erase(s) \in \sigma$.

Copycat property

We now prove the following:

$$\text{if } s m_1[m'_1] m_2[m'_2] \in \tilde{\sigma} \text{ with } \mathcal{A}(m_1), \mathcal{A}(m_2) \in \mathcal{O}_{A \to B}, \text{ then } erase(m'_1) = erase(m'_2)$$

We call it the **copycat property**. Note that this property will hold only because (σ, τ) is a game isomorphism, it is not true in general.

First consider the case where $S = sm_1[m'_1]m_2[m'_2]$ is symbolic. We note $u = erase(S)$ and we have $u \in \sigma$ and $v = \breve{u} \in \tau$. We will prove by recurrence that $erase(m'_1) = erase(m'_2)$, and that it is possible to build a play $T \in \tilde{\tau}$ such that $erase(T)$ is a copycat extension of v. So, suppose this is true for every $t \in \mathbb{E}$ such that $t \leq s$.

We set $\begin{cases} x_1 = erase(m_1) \\ x_2 = erase(m_2) \end{cases}$, $\begin{cases} a_1 = \mathcal{A}(m_1) \\ a_2 = \mathcal{A}(m_2) \end{cases}$, $\begin{cases} x'_1 = erase(m'_1) \\ x'_2 = erase(m'_2) \end{cases}$ and $\begin{cases} a'_1 = \mathcal{A}(m'_1) \\ b'_2 = \mathcal{A}(m'_2) \end{cases}$

We have three cases:

- if $paux_{A\to B}(a_1)$ is undefined (so that $m'_1 = 0$), suppose $m'_2 \neq 0$: then $paux_{A\to B}(a_2) = \mathbf{P}$ (because $\sharp(m'_2) = 0$ and $\mathbf{O}$ has only played symbolically), so $paux_{B\to A}(a_2) = \mathbf{O}$. As we have $T' \in \tilde{\tau}$ such that $t' = erase(T')$ is a copycat extension of t (so $t' \in \tau$ by hyperuniformity), one can build $T'm_2[j] \in \mathcal{P}_{B\to A}$ for some $j \neq 0$ (remember $\mathbf{O}$ plays symbolically), so by definition of the realization there exists $N \in \mathcal{M}_{B\to A}$ such that $T'm_2[j]N \in \tilde{\tau}$. Then $t'x_2[j]y \in \tau$ with $y = erase(N)$ and $\sharp(y) = j$, so by hyperuniformity $t'x_2[x'_2]y[x'_2] \in \tau$: this breaks determinacy since $tx_2[x'_2]x_1[0] \in \tau$ implies $t'x_2[x'_2]x_1[0] \in \tau$ by hyperuniformity, so $m'_2 \neq 0$ is impossible. Finally, we have $T'' = T'm_2[0]m_1[0] \in \tilde{\tau}$ with $erase(T'')$ copycat extension of v.

- if $paux_{A\to B}(a_1) = \mathbf{O}$, then $sm_1[j]m_2[M[j]] \in \tilde{\sigma}$ for some j such that $X_j \notin FTV(A)$, so $paux_{A\to B}(a_2)$ is defined. The case $paux_{A\to B}(a_2) = \mathbf{P}$ implies $paux_{B\to A}(a_2) = \mathbf{O}$; as we have $T' \in \tilde{\tau}$ such that $t' = erase(T')$ copycat extension of t (so $t' \in \tau$ by hyperuniformity), there exists k such that $T'm_2[k] \in \mathcal{P}_{B\to A}$ so $T'm_2[k]N[N'] \in \tilde{\tau}$ for some typed moves N, N' with $\mathcal{A}(N) \in \mathcal{O}_{B\to A}$. Hence $t'x_2[k]y[y'[k]] \in \tau$ with $y = erase(N)$ and $y' = erase(N')$, and by hyperuniformity $t'x_2[x'_2]y[y'[x'_2]] \in \tau$. But $tx_2[x'_2]x_1[j] \in \tau$ implies $t'x_2[x'_2]x_1[j] \in \tau$ by hyperuniformity, so $y'[x'_2] = j$ (hence $x'_2 = j$) and $y[j] = x_1[j]$.

  The case $paux_{A\to B}(a_2) = \mathbf{O}$ directly implies $M[j] = j$. One still has to build in this case the play $T \in \tilde{\tau}$ such that $erase(T)$ is a copycat extension of v: we have $T' \in \tilde{\tau}$ such that $t' = erase(T')$ copycat extension of t (so $t' \in \tau$ by hyperuniformity); moreover, if $D = \frac{m_2}{\mathcal{L}_A(a_2)}$ then there is at least one initial move $M[j] \in \mathcal{M}_D$. So, $T'm_2[M[j]] \in \mathcal{P}_{B\to A}$, and then $T = T'm_2[M[j]]N[N'] \in \tilde{\tau}$ for some typed moves N, N' with $\mathcal{A}(N) \in \mathcal{O}_{B\to A}$. Hence $t'x_2[z]y[y'] \in \tau$ with $z = erase(M)$, $y = erase(N)$ and $y' = erase(N')$. But $tx_2[j]x_1[j] \in \tau$ implies $t'x_2[z]x_1[z]$ by hyperuniformity, so by determinacy $erase(T) = t'x_2[z]x_1[z]$: it is a copycat extension of v.

- if $paux_{A\to B}(a_1) = \mathbf{P}$, then $paux_{B\to A}(a_1) = \mathbf{O}$. As we have $T' \in \tilde{\tau}$ such that $t' = erase(T')$ copycat extension of t, one can build as above $T'm_2[M] \in \mathcal{P}_{B\to A}$ for some typed move M (if $D = \frac{m_2}{\mathcal{L}_A(a_2)}$ then there is at least one initial move $M \in \mathcal{M}_D$), so $T'm_2[M]N[N'] \in \tilde{\tau}$ for some N, N' with $\mathcal{A}(N) = c \in \mathcal{O}_{B\to A}$. We set $y = erase(M)$, $z = erase(N)$ and $z' = erase(N')$. There are two possibilities: if $paux_{A\to B}(a_2) = \mathbf{O}$, then $M = j$ for some j. As $tx_2[j]x_1[x'_1] \in \tau$, one has $t'x_2[y]x_1[x'_1[y]] \in \tau$ by hyperuniformity, so $x_1[x'_1[y]] = z[z']$ by determinacy. This means $z[j] = x_1[j]$, so $paux_{B\to A}(c) = \mathbf{O}$. Hence $z'[k] = k$, and so $y[k] = x'_1[k] = k$. If $paux_{A\to B}(a_2) = \mathbf{P}$, then $paux_{B\to A}(a_2) = \mathbf{O}$ so $y = j$ for some j. As $tx_2[k]z[z'[k]] \in \tau$, one has $t'x_2[x'_2]z[z'[x'_2]] \in \tau$ by hyperuniformity, so $z[z'[x'_2]] = x_1[x'_1]$ by determinacy. This means $z[k] = x_1[k]$, so $paux_{B\to A}(c) = \mathbf{O}$. Then $z'[k] = k$, so $x'_1 = x'_2$.

Finally, if S is not symbolic, there exists a symbolic play $S' = s'M_1[M'_1]M_2[M'_2] \in \tilde{\sigma}$ whose S is a copycat extension. So $erase(M'_1) = erase(M'_2)$ and $erase(m'_1) = erase(m'_2)$ because of the definition of the copycat extension.

Construction of the untyped copycat play

Let a be a node of $\mathcal{F}_A$ and $a_1, \dots, a_p$ be the sequence of nodes of $\mathcal{F}_A$ such that $\vdash a_1$, $a_i \vdash a_{i+1}$ and $a_p = a$. We are going to construct a function $f : \mathcal{F}_A \to \mathcal{F}_B$ such that, for any $i \in \mathbb{N}$:

$$\mathcal{E}'(f(a_1))[i]\mathcal{E}'(a_1)[i]\mathcal{E}'(a_2)[i]\mathcal{E}'(f(a_2))[i]\mathcal{E}'(f(a_3))[i]\mathcal{E}'(a_3)[i]\cdots \in \sigma$$

where $\mathcal{E}' = \mathcal{E} \circ or : \mathcal{F}_A \cup \mathcal{F}_B \to \mathbb{X}$.

The construction of f will use the determinacy of σ and τ, which generates a unique move starting from a play with its complete history (not just the last move). That is why we could not work with a function $f' : \mathcal{O}_A \to \mathcal{O}_B$, because in that case, the choice of $f'(a)$ would depend not only on a, but also on the choice of the ancestors. As said at the end of section 7, the information contained in a node $c \in \mathcal{F}_A$ is precisely the node $or(c) \in \mathcal{O}_A$ plus its ancestors: so, the forests are the good structure to ensure that the function f is well-defined. Having this in mind, one can identify $or(a_i)$ (resp. $or(f(a_i))$) with a_i (resp. $f(a_i)$), and try to prove: $\mathcal{E}(f(a_1))[i]\mathcal{E}(a_1)[i]\mathcal{E}(a_2)[i]\mathcal{E}(f(a_2))[i]\mathcal{E}(f(a_3))[i]\mathcal{E}(a_3)[i]\cdots \in \sigma$.

Moreover, by the property of non-ambiguity (cf. def. 11), one has, for any b in $\mathcal{F}_A$ (resp. in $\mathcal{F}_B$): $\mathcal{E}(a_i) = \mathcal{E}(b) \Rightarrow b = a_i$ (resp. $\mathcal{E}(f(a_i)) = \mathcal{E}(b) \Rightarrow b = f(a_i)$). That is why we will also identify a_i (resp. $f(a_i)$) with $\mathcal{E}(a_i)$ (resp. $\mathcal{E}(f(a_i))$). What has to be proved is then:

$$f(a_1)[i]a_1[i]a_2[i]f(a_2)[i]f(a_3)[i]a_3[i]\cdots \in \sigma$$

If $p = 1$, we build a symbolic play $s = m_1 \in \mathcal{P}_{B\to A}$ such that $erase(m_1) = a_1[i]$ for $X_i = \frac{m_1}{\mathcal{L}_{B\to A}(a_1)}$. Let b be the only untyped move such that $a_1[i]b \in \tau$ (which exists by totality of τ). $\tilde{\tau}$ being a realisation of τ, there must be a play $m_1 M \in \tilde{\tau}$ with $erase(M) = b$, and we have a decomposition $M = m_2[m_2']$ with $\mathcal{A}(m_2) = c \in \mathcal{O}_{B\to A}$. We choose $f(a_1) = c$, and we have $erase(m_2') = i$ because of the copycat property.

If $p = p' + 1$ with p' odd, we have by induction hypothesis: $f(a_1)[i]a_1[i]a_2[i]f(a_2)[i]\dots f(a_{p'})[i]a_{p'}[i] \in \sigma$, and by totality of σ there exists a unique move x such that $f(a_1)[i]a_1[i]a_2[i]f(a_2)[i]\dots f(a_{p'})[i]a_{p'}[i]a_p[i]x \in \sigma$. One is able to build inductively a play $S \in \sigma$ such that $erase(S) = f(a_1)[y_1]a_1[y_1]\dots f(a_{p'})[y_{p'}]a_{p'}[y_{p'}]$ for a good choice of the moves y_k: indeed, if $S' \in \sigma$ with $erase(S') = f(a_1)[y_1]a_1[y_1]\dots f(a_k)[y_k]a_k[y_k]$ (the case $erase(S') = f(a_1)[y_1]a_1[y_1]\dots a_k[y_k]f(a_k)[y_k]$ is similar), we choose a move $M = m_1[m_2]$ with $\mathcal{A}(m_1) = a_{k+1}$ and m_2 initial move of $\mathcal{M}_D$ where $D = \frac{m_1}{\mathcal{L}_{A\to B}(a_{k+1})}$; we note $erase(m_{k+1}) = y_{k+1}$. $\tilde{\sigma}$ being a realisation of σ, we have $S'MM' \in \tilde{\sigma}$ for some typed move M', and $erase(M') = a_{k+1}[y_{k+1}]$ by hyperuniformity of σ. Hence we have obtained $S = S'MM' \in \tilde{\sigma}$ such that $erase(S) = f(a_1)[y_1]a_1[y_1]\dots a_{k+1}[y_{k+1}]f(a_{k+1})[y_{k+1}]$ for some y_{k+1} [7].

Then one chooses a typed move N such that $tN \in \mathcal{P}_{A\to B}$ and $erase(M) = a_p[y_p]$ for some initial move y_p (it suffices once again to choose y_p as initial in $\mathcal{M}_D$ for the appropriate game D).

As $f(a_1)[y_1]a_1[y_1]\dots f(a_{p'})[y_{p'}]a_{p'}[y_{p'}]a_p[y_p]x[y_p] \in \sigma$ by hyperuniformity, we have $tNN' \in \tilde{\sigma}$ for some N' with $erase(N') = x[y_p]$. So $x = b[z]$ with $b \in \mathcal{O}_{A\to B}$, and we choose $f(a_p) = b$. By the copycat property $z[y_p] = y_p$, so $z[i] = i$: this means $f(a_1)[i]a_1[i]a_2[i]f(a_2)[i]\dots a_p[i]f(a_p)[i] \in \sigma$.

If $p = p' + 1$ with p' even, one can do the same reasoning by using τ instead of σ.

In the same way, one can associate a function g with τ and easily verify that $f \circ g$ is the identity on $\mathcal{O}_B$ and $g \circ f$ is the identity on $\mathcal{O}_A$, so f is a bijection. Moreover, by construction, if $a \le a'$ then $f(a) \le f(a')$.

Construction of the typed copycat play

To prove that f satisfies the requirements of a Curry-isomorphism, we will construct a play $s_p \in \tilde{\sigma}$ such that $erase(s_p) = t_p$, where $t_p = \begin{cases} f(a_1)[y_1]a_1[y_1]a_2[y_2]f(a_2)[y_2]\dots f(a_p)[y_p]a_p[y_p] & \text{if } p \text{ odd} \\ f(a_1)[y_1]a_1[y_1]a_2[y_2]f(a_2)[y_2]\dots a_p[y_p]f(a_p)[y_p] & \text{if } p \text{ even} \end{cases}$
for an appropriate choice of the moves y_i. Moreover, one will have $s_p = s_{p-1}m_1[m_2]M'$ where $\mathcal{A}(m_1) = c \in \mathcal{O}_{A\to B}$ and $erase(m_2) = y_p$ uniquely determined by $D = \frac{m_1}{\mathcal{L}_{A\to B}(c)}$.

In the plays s_p, we will use the games $(C_j)_{j\in\mathbb{N}}$ defined by: $C_1 = \bot \times \bot$ and $C_{j+1} = C_j \times C_j$. Note that each initial move of C_j takes the form $b_1(b_2(\dots(b_j(0))\dots))$, where each b_i can be either r or l. We call r_j the initial move of C_j where each b_i is equal to r. These games will be used in order to have "fresh" moves, i.e. moves that cannot come from a game defined before C_j is played. In what follows, the integer n_p is made to ensure that no game defined before step p can belong to C_q for $q \ge p$.

We now build the triple (s_p, y_p, n_p) inductively:

[7] In the next part of the proof we will also build a typed play $s_p \in \tilde{\sigma}$ such that $erase(s_p) = f(a_1)[y_1]a_1[y_1]\dots f(a_p)[y_p]a_p[y_p]$, but there will be more constraints on s_p.

- If $p = 1$, we define the typed move $M_1 = m_1[m_2]$ such that: $\mathscr{A}(m_1) = f(a_1)$, the d_1 games played at the level of m_1 are $C_1, \ldots, C_{d_1}$ and we choose $m_2 = \sharp(f(a_1))$ if $paux_{A\to B}(f(a_1))$ is undefined, $m_2 = r_j$ if $\frac{m_1}{\mathcal{L}_{A\to B}(f(a_1))} = C_j$. As $M_1 \in \mathcal{P}_{A\to B}$ and $\tilde{\sigma}$ is a realization of σ, there exists M'_1 such that $M_1M'_1 \in \tilde{\sigma}$ and $erase(M_1M'_1) = f(a_1)[y_1]a_1[y_1]$, where $y_1 = erase(m_2)$. We choose $s_1 = M_1M'_1$. Let us define N as the biggest number of tokens r in any initial occurrence of a game D defined at the level of M'_1. We choose $n_1 = max(d_1, N) + 1$.
- If $p = p' + 1$ with p' odd, we define the typed move $M_p = m_1[m_2]$ such that: $erase(m_1) = f(a_p)$, the d_p games defined at the level of m_1 are $C_{n_{p'}}, \ldots, C_{n_{p'}+d_p}$ and m_2 is chosen as follows:
  - if $paux_{A\to B}(f(a_p))$ is undefined, $m_2 = \sharp(f(a_p))$
  - if $paux_{A\to B}(f(a_p)) = \mathbf{O}$, $m_2 = r_j$ if $\frac{m_1}{\mathcal{L}_{A\to B}(f(a_p))} = C_j$
  - if $paux_{A\to B}(f(a_p)) = \mathbf{P}$, let $D = \frac{m_1}{\mathcal{L}_{A\to B}(f(a_p))}$. Either there exists $c \in \mathcal{O}_D$ such that $\vdash c$ and $\mathcal{L}_D(c) \neq \dagger$, and in this case one chooses $m_2 = m'_2[m'_3]$ with $\mathscr{A}(m'_2) = c$, $\frac{m'_2}{\mathcal{L}_D(c)} = \bot \times \bot$ and $m'_3 = l0$; we note $r_D = erase(m_2)$ [8], and we require that r_D is a function of D; or there exists no such c and in this case we choose m_2 such that $\mathscr{A}(m_2)$ is one of the initial occurrences of D: we just require that this choice is a function of D, and note it r_D.

  As $s_{p'}M_p \in \mathcal{P}_{A\to B}$ and $\tilde{\sigma}$ is a realization of σ, there exists M'_p such that $s_{p'}M_pM'_p \in \tilde{\sigma}$ and $erase(s_{p'}M_pM'_p) = t_p$ if $y_p = erase(m_2)$. We choose $s_p = s_{p'}M_pM'_p$. Let us define N as the biggest number of tokens r in any initial occurrence of a game D defined at the level of M'_p. We choose $n_p = max(n_{p'} + d_p, N) + 1$.
- If $p = p' + 1$ with p' odd, we do the same choices as in the preceding case, except that $f(a_p)$ must be replaced by a_p, and conversely.

Suppose $paux_{A\to B}(a_p)$ is defined, then $ref_{A\to B}(a_p) = b$ is also defined. It is important for the next section of the proof to understand the link between b and the play s_p. First, note that $b = a_i$ for some $1 \leq i \leq p$; then, because of the definition of the set $\mathcal{R}_{A\to B}$ of hyperedges, we know that a_i is the minimal occurrence c of $\mathcal{O}_{A\to B}$ such that $\mathcal{L}_{A\to B}(a_p)$ is a prefix of c. Hence, if M_i (resp. M_p) is the move in s_p such that $erase(M_i) = a_i[y_i]$ (resp. $erase(M_p) = a_p[y_p]$) and if $D = \frac{M_p}{\mathcal{L}_{A\to B}(a_p)}$, then the game D is played by $paux_{A\to B}(a_p)$ at the level of M_i. So, in the construction of s_p, D has been played at step i.

We also need to build a play $u_p \in \tilde{\tau}$ such that $erase(u_p) = v_p$, where $v_p = \begin{cases} a_1[y'_1]f(a_1)[y'_1]f(a_2)[y'_2]a_2[y'_2]\ldots a_{p'}[y_{p'}]f(a_{p'})[y_{p'}] & \text{if } p \text{ odd} \\ a_1[y'_1]f(a_1)[y'_1]f(a_2)[y'_2]a_2[y'_2]\ldots f(a_p)[y'_p]a_p[y'_p] & \text{if } p \text{ even} \end{cases}$ for an appropriate choice of the moves y'_i.
The procedure is similar (we just need to swap σ and τ). Note that we do not have in general $u_p = \check{s}_p$, or even $erase(u_p) = \check{w}_p$ with $w_p = erase(s_p)$, because the untyped moves y_i and y'_i may differ.

Curry-isomorphism

We are now going to prove that the bijection f satisfies each requirement of a Curry-isomorphism.

We first prove that $\mathcal{D}_B \circ f = \mathcal{D}_A$: suppose $\mathcal{D}_A(a_p) = X_i$, then $s_p = s_{p-1}MM'$ with $erase(M) = a_p[i]$ and $erase(M') = f(a_p)[i]$; likewise, $u_p = u_{p-1}NN'$ with $erase(N) = f(a_p)[i]$ and $erase(M') = a_p[i]$. If $paux_{A\to B}(f(a_p)) = \mathbf{O}$ then one should have $i = r_j$ for some j by construction of s_p, which is impossible. If $paux_{A\to B}(f(a_p)) = \mathbf{P}$ then $paux_{B\to A}(f(a_p)) = \mathbf{O}$ and one should have $i = r_j$ for some j by construction of u_p, which is impossible. Then $paux_{A\to B}(f(a_p))$ is not defined, and $\sharp(f(a_p)) = i$ which means $\mathcal{D}_B(f(a_p)) = X_i$. Similarly, $\mathcal{D}_B(f(a_p)) = X_i$ implies $\mathcal{D}_A(a_p) = X_i$ as well.

[8] In this case, m_2 is precisely built in such a way that we cannot have $r_D = r_j$ for any j.

We then prove that $f(\mathcal{S}_A) = \mathcal{S}_B$: if $a_p \in S$ with $(t, S) \in \mathcal{R}_A$ for some t, suppose $\mathcal{L}_{A\to B}(f(a_p)) = \dagger$. If $paux_{A\to B}(a_p) = \mathbf{O}$ then $s_p = s_{p-1}MM'$ with $erase(M) = a_p[y_p]$ and $erase(M') = f(a_p)[y_p]$, and one should have $y_p = r_j$ for some j by construction of s_p. But this is impossible since $\mathcal{L}_{A\to B}(f(a_p)) = \dagger$ implies $\mathcal{A}(M') \in \mathcal{O}_{A\to B}$, so $y_p \in \mathbb{N}$. If $paux_{A\to B}(a_p) = \mathbf{P}$ then $paux_{B\to A}(a_p) = \mathbf{O}$, $u_p = u_{p-1}NN'$ with $erase(N) = f(a_p)[y_p]$ and $erase(N') = a_p[y_p]$ and one should have $y_p = r_j$ for some j by construction of u_p. But this is impossible since $\mathcal{L}_{A\to B}(f(a_p)) = \dagger$ implies $\mathcal{A}(N) \in \mathcal{O}_{A\to B}$, so $y_p \in \mathbb{N}$.

Finally, we need to prove the following: for every $(t, S) \in \mathcal{R}_A$, if there exists $c \in S$ such that $\lambda(c) \neq \lambda(t)$, then $(f(t), f(S)) \in \mathcal{R}_B$ (the reciprocal would be done similarly). Let us take $a_1, \dots, a_p$ the sequence of nodes such that: $\vdash a_1$, $a_i \vdash a_{i+1}$ and $a_p = c$. We necessarily have $t = a_i$ for some $i \leq p$.

First we prove that $ref_B(f(a_p)) = f(a_i)$: suppose that it is false, then $ref_B(f(a_p)) = f(a_j)$ with $j \neq i$. First take $j < i$: if $paux_{A\to B}(a_p) = \mathbf{O}$, then $f(a_p)$ is an $\mathbf{O}$-move on $A \to B$, so $s_p = SM_pM'_p$ where: $M_p = m_1[m_2]$, $\mathcal{A}(m_1) = f(a_p)$ and $\frac{m_1}{\mathcal{L}_{A\to B}(f(a_p))} = D$ for some D chosen at step j; and $M'_p = m'_1[m'_2]$, $\mathcal{A}(m'_1) = a_p$ and $\frac{m'_1}{\mathcal{L}_{A\to B}(a_p)} = C_{k'}$ for some $k' \geq n_{i-1}$. So we should have $y_p = r_k$ to be the move we choose in D, which is impossible by construction of n_{i-1}. If $paux_{A\to B}(a_p) = \mathbf{P}$, we simply note that $paux_{B\to A}(a_p) = \mathbf{O}$ and do the same reasoning with u_p in $B \to A$. In the case where $i < j$, the reasoning is similar: if $paux_{A\to B}(a_p) = \mathbf{P}$, then $s_p = SM_pM'_p$ where: $M_p = m_1[m_2]$, $\mathcal{A}(m_1) = a_p$ and $\frac{m_1}{\mathcal{L}_{A\to B}(a_p)} = D$ for some D chosen at step i; and $M'_p = m'_1[m'_2]$, $\mathcal{A}(m'_1) = f(a_p)$ and $\frac{m'_1}{\mathcal{L}_{A\to B}(f(a_p))} = C_{k'}$ for some $k' \geq n_{j-1}$. This leads to a contradiction. If $paux_{A\to B}(f(a_p)) = \mathbf{P}$, we work on $B \to A$.

Let us now have $b \in fr_A(a_p)$, and suppose that $f(b) \notin fr_B f(a_p)$. By what has been proved before we know that $paux_{A\to B}(f(b))$ is defined, but also that $ref_B(f(b))$ has the same polarity as $ref_A(b)$: indeed, if $paux_A(b) \neq \lambda(b)$ then $ref_B(f(b)) = f(ref_A(b))$, so $\lambda(ref_B(f(b))) = \lambda(f(ref_A(b))) = \lambda(ref_A(b))$; similarly, if $paux_B(f(b)) \neq \lambda(f(b))$ then $ref_A(b) = f^{-1}(ref_B(f(b)))$, so $\lambda(ref_A(b)) = \lambda(f^{-1}(ref_B(f(b)))) = \lambda(ref_B(f(b)))$. Finally, if $paux_A(b) = \lambda(b)$ and $paux_B(f(b)) = \lambda(f(b))$ then $paux_A(b) = paux_B(f(b))$ because b and $f(b)$ have the same polarity. Then, in all cases, $paux(b) = paux(f(b))$.

We consider that $paux_{A\to B}(f(b)) = \mathbf{O}$ (if not, one works with u_p on $B \to A$), so $paux_{A\to B}(a_p) = \mathbf{P}$ and $s_p = s_{p-1}m_1[m_2]m'_1[m'_2]$ with $\frac{m'_1}{\mathcal{L}_{A\to B}(f(a_p))} = C_k$ for some k. Let $D = \frac{m_1}{\mathcal{L}_{A\to B}(a_p)}$, we necessarily have that $y_p = r_k = erase(r_D)$. But a problem arises with b and $f(b)$: as a first case, suppose that b has the polarity $\mathbf{P}$ in A. Then there is a play $s'_q = s'_{q-1}M_1[M_2]M'_1[M'_2]$ in $\tilde{\sigma}$ constructed the same way as s_p, such that $erase(s'_q) = erase(s'_{q-1})b[y'_q]f(b)[y'_q]$, and where $\frac{M_1}{\mathcal{L}_{A\to B}(b)} = D$ and $\frac{M'_1}{\mathcal{L}_{A\to B}(f(b))} = C_{k'}$ with $k' \neq k$. Then we should have $y'_q = erase(r_D)$ occurrence of $C_{k'}$, so $r_k = r_{k'}$ which is impossible.

The second case is where b has the polarity $\mathbf{O}$ in A. Then there is a play $s'_q = M_1[M_2]M'_1[M'_2]$ in $\tilde{\sigma}$ constructed the same way as s_p, such that $erase(s'_q) = s' f(b)[y'_q]b[y'_q]$, and where $\frac{M_1}{\mathcal{L}_{A\to B}(f(b))} = C_{k'}$ with $k' \neq k$ and $\frac{M'_1}{\mathcal{L}_{A\to B}(b)} = D$. Then we should have $y'_q = r_{k'} = erase(d')$ with d' move in D. But in this case $\mathcal{A}(d), \mathcal{A}(d') \in \mathcal{O}_D$ (if not we have a token l in d or d'), so $\mathcal{E}(d) = r_k$ and $\mathcal{E}(d') = r_{k'}$, hence $k = k'$ because D is unambiguous. This is impossible.

$f(b) \in fr_B(f(a_p))$ similarly implies $b \in fr_A(a_p)$, so $f(S) = \{b \mid s \in fr_B(f(a_p))\}$. This allows us to conclude that $(f(t), f(S)) \in \mathcal{R}_B$. ◻